



# Etude et conception d'algorithmes pour les réseaux mobiles et ad-hoc

Badis Hakim

## ► To cite this version:

Badis Hakim. Etude et conception d'algorithmes pour les réseaux mobiles et ad-hoc. Réseaux et télécommunications [cs.NI]. Université Paris Sud - Paris XI, 2005. Français. NNT: . tel-00637112

**HAL Id: tel-00637112**

**<https://theses.hal.science/tel-00637112>**

Submitted on 30 Oct 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : XXXX

THÈSE  
de  
L'UNIVERSITÉ PARIS-SUD

DOCTORAT DE L'UNIVERSITÉ PARIS-SUD  
Spécialité : INFORMATIQUE

Présentée et soutenue publiquement

Par

HAKIM BADIS

ÉTUDE ET CONCEPTION D'ALGORITHMES  
POUR LES RÉSEAUX MOBILES ET AD HOC

Directeur de thèse : M. Khaldoun Al Agha

Jury

Mme	Isabelle Guérin Lassous	Professeur	Rapporteur
M.	David Symplot-Ryl	Professeur	Rapporteur
M.	Khaldoun Al Agha	Professeur	Examineur
M.	Philippe Jacquet	Directeur de Recherche I.N.R.I.A.	Examineur
M.	Guy Pujolle	Professeur	Examineur
M.	xxxxxx	xxxxxx	xxxxxx

Laboratoire de Recherche en Informatique, U.M.R. CNRS 8623,  
Université Paris-Sud, 91405 Orsay Cedex, France





# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Le point de départ et sujet de thèse . . . . .	5
1.2	Contributions et organisation du document . . . . .	7
<b>2</b>	<b>La mobilité dans les réseaux sans fil : état de l'art</b>	<b>11</b>
2.1	Historique, évolution et applications des réseaux sans fil . . . . .	11
2.2	Problèmes des transmissions radios et des réseaux sans fil . . . . .	13
2.3	Les réseaux mobiles et sans fil . . . . .	14
2.3.1	Les réseaux cellulaires . . . . .	16
2.3.2	IP Mobile . . . . .	17
2.3.3	Les réseaux privés sans fil . . . . .	19
2.4	Résumé . . . . .	24
2.5	Conclusion . . . . .	25
<b>3</b>	<b>La 4G : l'interopérabilité entre les réseaux</b>	<b>27</b>
3.1	Les principes de la gestion de mobilité dans les réseaux sans fil et mobiles . . . . .	27
3.2	Une structure hiérarchique des réseaux sans fil . . . . .	28
3.3	Un exemple d'intégration : 3G/WLAN . . . . .	30
3.4	Les handovers dans une OWN . . . . .	31
3.5	Position des objectifs . . . . .	32
3.6	OWN : La gestion de la mobilité . . . . .	33
3.6.1	Description d'Archi-1 . . . . .	34
3.6.2	Fonctionnement d'Archi-1 . . . . .	35
3.6.3	Les handovers dans Archi-1 . . . . .	36
3.6.4	Latence et overhead dans Archi-1 . . . . .	37
3.6.5	Description d'Archi-2 . . . . .	38
3.6.6	Fonctionnement d'Archi-2 . . . . .	39
3.6.7	Les handovers dans Archi-2 . . . . .	39
3.6.8	Latence et Overhead dans Archi-2 . . . . .	41
3.7	Optimisation des handovers . . . . .	41
3.7.1	La cellule virtuelle . . . . .	42
3.7.2	Émission rapide des trames balises . . . . .	43
3.7.3	Diffusion multiple des paquets (DMP) . . . . .	43
3.7.4	Diffusion multiple des entêtes (DFE) . . . . .	44
3.8	Les handovers inter-domaines . . . . .	45
3.9	La mobilité rapide et faible . . . . .	45

3.10	Évaluation des performances . . . . .	46
3.10.1	Le modèle de simulation . . . . .	46
3.10.2	Les résultats des simulations . . . . .	49
3.11	Conclusion . . . . .	53
<b>4</b>	<b>Conception des algorithmes pour le routage à multiples contraintes</b>	<b>55</b>
4.1	Interconnexions et graphes . . . . .	56
4.2	Les règles de composition des contraintes . . . . .	57
4.3	Résumé des algorithmes de routage existants . . . . .	58
4.4	Routage à contrainte simple . . . . .	60
4.5	Routage aux contraintes multiples . . . . .	60
4.5.1	Une approche à métrique unique mixte . . . . .	61
4.5.2	Une approche à plusieurs métriques . . . . .	61
4.6	Résultats de simulation . . . . .	78
4.6.1	Le coût moyen . . . . .	78
4.6.2	Le nombre moyen d'étapes . . . . .	79
4.6.3	Comparaison des performances . . . . .	80
4.7	Conclusion . . . . .	81
<b>5</b>	<b>Routage ad hoc et qualité de service</b>	<b>83</b>
5.1	Les réseaux Ad hoc . . . . .	83
5.1.1	Routage niveau MAC vs niveau réseau . . . . .	84
5.1.2	Les protocoles de routage pour les réseaux Ad Hoc . . . . .	84
5.2	Optimized Link State Routing protocol (OLSR) . . . . .	86
5.2.1	Détection de voisinage . . . . .	86
5.2.2	Le concept des relais multipoint . . . . .	87
5.2.3	Gestion de la topologie . . . . .	88
5.2.4	Calcul des routes . . . . .	89
5.2.5	Raisons de choix du protocole OLSR . . . . .	89
5.3	Définition de la qualité de service . . . . .	90
5.4	Les concepts de QoS sur IP . . . . .	91
5.4.1	IntServ et DiffServ . . . . .	91
5.4.2	RSVP . . . . .	92
5.4.3	Routage contraint . . . . .	93
5.5	Les modèles de QoS pour MANETs . . . . .	94
5.5.1	FQMM : Flexible quality of service model for MANETs . . . . .	94
5.5.2	SWAN : Service differentiation in wireless ad hoc networks . . . . .	95
5.5.3	Modèle iMAQ . . . . .	96
5.6	Systèmes de signalisation pour la QoS dans MANETs . . . . .	96
5.6.1	Signalisation in-band et out-of-band . . . . .	96
5.6.2	Maintient des réservations soft-state et hard-state . . . . .	97
5.6.3	Protocole de signalisation INSIGNIA . . . . .	97
5.6.4	Le protocole Bruit . . . . .	99
5.7	Routage avec QoS dans MANETs . . . . .	99
5.7.1	CEDAR : Core-Extraction Distributed Ad hoc Routing Algorithm . . . . .	100
5.7.2	Ticket-Based QoS Routing . . . . .	100
5.7.3	QoS pour AODV . . . . .	101

5.8	Conclusion . . . . .	102
<b>6</b>	<b>EQMM : Efficient Quality of service Model for MANETs</b>	<b>103</b>
6.1	Description du modèle . . . . .	103
6.2	OLSR : QoS implicite . . . . .	105
6.3	Le protocole de routage QOLSR . . . . .	105
6.3.1	Principe de fonctionnement de QOLSR . . . . .	106
6.3.2	La détection de voisinage et mesure des métriques . . . . .	107
6.3.3	La sélection des relais multipoints (MPRs) . . . . .	108
6.3.4	La sélection des MPRs-1 . . . . .	110
6.3.5	La sélection des MPRs-2 . . . . .	111
6.3.6	La déclaration de la topologie et les conditions de QoS . . . . .	114
6.3.7	Calcul des routes . . . . .	115
6.3.8	Oscillation des flux . . . . .	115
6.4	Estimation des métriques . . . . .	116
6.4.1	Une méthode analytique pour le calcul du délai . . . . .	116
6.4.2	Une méthode analytique pour le calcul de la bande passante . . . . .	124
6.4.3	Une méthode expérimentale pour le calcul du délai . . . . .	126
6.4.4	Une méthode expérimentale pour le calcul de la bande passante . . . . .	127
6.5	Contrôle d'admission et de réservation . . . . .	136
6.5.1	Maintenance de la réservation . . . . .	139
6.5.2	La déconnexion des routes . . . . .	139
6.5.3	Transmission de la signalisation . . . . .	139
6.6	Contrôle de congestion . . . . .	139
6.7	QOLSR : Évaluation des performances . . . . .	140
6.7.1	Les hypothèses de la simulation . . . . .	140
6.7.2	Architecture générale . . . . .	141
6.7.3	Modèle de simulation . . . . .	141
6.7.4	Résultats de simulation . . . . .	143
6.8	Conclusions . . . . .	153
<b>7</b>	<b>Conclusions et perspectives</b>	<b>155</b>
7.1	Conclusion . . . . .	155
7.2	Perspectives . . . . .	157
<b>A</b>	<b>Spécification des paquets de contrôle de QOLSR</b>	<b>159</b>
A.1	Le nouveau format d'un message <i>Hello</i> . . . . .	159
A.2	Le nouveau format d'un message <i>TC</i> . . . . .	160
A.3	Le codage TVL . . . . .	160
<b>B</b>	<b>Liste de publications</b>	<b>163</b>



# Chapitre 1

## Introduction

### 1.1 Le point de départ et sujet de thèse

Les systèmes de télécommunication ont subi en l'espace de deux décennies des évolutions et bouleversements profonds. Dans le panorama de ces systèmes, les réseaux mobiles occupent, notamment depuis la fin des années 1980, une place chaque jour plus grande en termes de recherche, d'investissements, de revenus, d'abonnés, etc. Il est indéniable que l'association des services mobiles aux communications sans fil est l'une des plus importantes évolutions qu'ait observé le secteur des télécommunications depuis le début des années 1990. Ainsi l'Internet mobile est devenu peu à peu une réalité pour le grand public : au bureau avec les réseaux sans fil, dans les lieux publics avec les hotspots Wi-Fi (*Wireless Fidelity*), en déplacement grâce au GPRS (*General Packet Radio Services*) et les réseaux de 3ème génération. Cette réalité a un impact fondamental dans les modes de vie et de travail d'une partie croissante de la population. Selon le cabinet International Data Corporation (*IDC*), dans une étude sur la mobilité parue en décembre 2002 [1], plus de 4,5 millions de personnes en France sont déjà mobiles au sein de leur entreprise, et plus de 6 millions le sont à l'extérieur. Cette étude a souligné que le retard pris par le GPRS et l'UMTS (*Universal Mobile Telecommunications System*), a permis aux réseaux locaux sans fil (*WLAN : Wireless Local Area Network*) de se développer rapidement sur trois secteurs. D'abord dans les secteurs de la distribution, de la santé (hôpitaux) et de l'éducation ; Puis dans le monde de l'entreprise de façon plus large, dans les lieux publics enfin (hôtels, aéroports, salles de colloques, etc). *IDC* prévoit qu'en 2005, 3 millions de portables pourraient être équipés d'une solution de type *WLAN*, représentant ainsi une alternative partielle au GPRS et à l'UMTS.

Avec les progrès réalisés dans les réseaux et *WLAN*, chercheurs et industriels ont déjà commencé le développement des futurs réseaux mobiles de 4ème génération (4G). Dans cette génération, l'équipement terminal de l'utilisateur recherchera en permanence le meilleur réseau d'accès en fonction des besoins de l'utilisateur. Ce dernier aura plusieurs technologies d'accès sans fil à sa disposition et souhaitera être connecté au mieux, n'importe où, n'importe quand et avec n'importe quel réseau d'accès. Pour cela, les différentes technologies sans fil qui seront présentes dans la 4G (UMTS, *WLAN*, *WMAN*, etc), doivent coexister de manière à ce que la meilleure technologie puisse être retenue en fonction du profil de l'utilisateur, du type de réseau d'accès, du terminal et du type d'application demandée. Parmi les premiers à avoir réalisé des tests et des maquettes de réseaux 4G, citons l'opérateur japonais NTT DoCoMo, l'équipementier Alcatel (centre de recherche de Shanghai) ou le fabricant sud-coréen Samsung. La commercialisation de services fondés sur ce nouveau réseau n'est cependant pas prévue avant trois à cinq ans en Asie, voire dix ou quinze ans



en Europe et aux Etats-Unis (lire [2]).

L'intégration de plusieurs technologies sans fil pose beaucoup de problèmes. On trouve parmi eux : l'Authentification, l'Autorisation et l'Accounting (AAA), la sécurité, la Qualité de Service (QoS) et les handovers. Un des objectifs de cette thèse est de proposer et analyser des protocoles de handovers entre les différents systèmes. Le changement de réseau doit être rapide et sans perte de données afin de garantir la transparence vis-à-vis des utilisateurs mobiles. Ces technologies peuvent aussi donner aux utilisateurs expérimentés la possibilité de sélectionner un réseau ou effectuer un handover.

L'infrastructure fixe joue un rôle important dans l'ensemble des réseaux sans fil notamment les réseaux 4G. Les réseaux ad hoc sont considérés comme généralisation ultime des réseaux sans fil car ils limitent au maximum le rôle de l'infrastructure fixe. En effet, ils ont pour but de connecter des entités communicantes (qui pourront être mobiles) en dehors de toute infrastructure pré-existante au sein de réseaux spontanés.

Le domaine militaire est par excellence le domaine de prédilection pour les réseaux ad hoc. Il adopte une recherche plus intensive pour obtenir les meilleures performances. Un réseau ad hoc doit pouvoir être déployé à la demande, fonctionner sans infrastructure de communication préexistante et bien sûr tolérer la mobilité. Il doit aussi garantir des échanges fiables et de qualité car de plus en plus les informations échangées sur le champ de bataille comportent des images et des vidéos. Il existe en Europe de nombreux programmes militaires qui utilisent la technologie des réseaux ad hoc : le programme FÉLIN (Fantassin à Équipement et Liaisons INTégrées) qui est un programme de la DGA (Délégation Générale de l'Armement), le programme FIST qui est le programme équivalent au Royaume-Uni et le programme RHD (Radio Haut Débit) qui est un programme français de conception d'une radio haut débit pour les réseaux ad hoc.

La recherche sur les réseaux ad hoc dans le domaine civil s'est intensifiée avec l'arrivée des technologies radio publiques, principalement la norme IEEE 802.11 et ses diverses extensions. La norme 802.11 est conçue avant tout pour former des réseaux autour d'une infrastructure fixe de bornes radio reliées entre elles par un réseau câblé. Mais la méthode d'accès au canal qu'elle propose par défaut fonctionne de manière totalement distribuée, c'est la Distributed Coordination Function (DCF). Cette dernière, alliée au coût modéré des équipements, a joué un rôle primordial pour l'utilisation de 802.11 dans un contexte ad hoc.

Puisque la propagation radio des entités d'un réseau ad hoc est limitée, un protocole de routage est nécessaire afin de faire communiquer deux entités quelconques dans le réseau. Le groupe de travail MANET (Mobile Ad hoc NETwork) [3] créé au sein de l'IETF (*Internet Engineering Task Force*) [4] a pour but de spécifier et de standardiser des protocoles de routage pour les réseaux ad hoc au niveau IP. Ces protocoles doivent pouvoir supporter des couches physique et MAC hétérogènes et offrent aux couches supérieures la possibilité de fonctionner uniquement sur IP.

Les protocoles de routage proposés par MANET sont conçus pour assurer uniquement la connectivité selon une seule métrique comme le nombre de sauts. Ils mettent en œuvre quelques éléments implicites pour assurer une meilleure qualité des liens et chemins. On trouve parmi eux : l'utilisation d'un plus court chemin, l'utilisation d'un nombre limité de nœuds relais, vérification de la qualité des liens, etc. Cependant, ces protocoles n'offrent aucun mécanisme de QoS (contrôle d'admission, routage contraints, contrôle de congestion, etc) et restent insuffisants pour satisfaire les flux qui exigent certaines contraintes.

Ces dernières années, on observe des progrès significatifs pour étudier la QoS dans les réseaux ad hoc mobiles. Il existe maintenant des travaux intéressants qui peuvent être classifiés en modèles de QoS, protocoles de routage contraints, protocoles de signalisation, extensions MAC pour la QoS, etc. Aucun de ces travaux n'est suffisant pour garantir une QoS de bout-en-bout à

tout moment. Certains d'entre eux sont des modèles théoriques permettant de comprendre le comportement du réseau avec la QoS. D'autres se basent sur des hypothèses plus au moins fortes. Cette thèse présente une vision critique des travaux intéressants dans le domaine de la QoS dans les réseaux ad hoc. Les points forts et faibles de chaque proposition sont exploités pour réaliser un modèle complet de QoS. Ce modèle possède un protocole de routage avec QoS qui trouve la meilleure route pour chaque type de flux selon la demande. Il limite l'apparition des congestions et la surutilisation de la bande passante par un mécanisme de contrôle d'admission et de réservation.

## 1.2 Contributions et organisation du document

Nos travaux réalisés dans le cadre de cette thèse consistent à proposer des algorithmes pour les handovers dans les réseaux 4G d'une part et des protocoles et mécanismes de QoS dans les réseaux ad hoc d'autre part. Le chapitre 2 récapitule les différentes technologies sans fil et mobiles. Elles sont classifiées selon l'étendue géographique et la topologie. Deux concepts sont présentés : le concept cellulaire qui consiste à diviser une zone de couverture relativement grande en plusieurs zones plus petites appelées cellules et celui des réseaux locaux sans fil qui offrent des services comparables à ceux des réseaux filaires dans des zones géographiquement limitées. La norme IEEE 802.11 est bien détaillée afin de l'utiliser par la suite dans les réseaux ad hoc. Le chapitre 5 introduit les réseaux ad hoc ainsi que les mécanismes de QoS existants.

Notre contribution se présente ensuite sous la forme de deux axes différents : le handover vertical dans les réseaux 4G (chapitre 3) et la QoS dans les réseaux ad hoc (chapitres 4 et 6).

Pour le handover vertical, le démarche suivie est :

- **Conception d'une structure hiérarchique des réseaux sans fil et mobiles** : cette structure est conçue en se basant sur une étude comparative entre les caractéristiques des systèmes existants comme la zone de couverture et la bande passante offerte. Le but de cette structure est de simplifier la gestion de la mobilité entre les systèmes hétérogènes. Le mobile dispose de plusieurs interfaces de technologies sans fil. Il peut basculer d'une technologie à une autre selon sa position dans cette structure. La structure proposée est virtuelle car nous considérons que le réseau de technologie *x* est toujours enveloppé par un autre de technologie *y*. En réalité, les deux réseaux *x* et *y* peuvent être totalement indépendants. L'ajout de composantes matérielles et/ou logicielles est nécessaire pour raccorder deux ou plusieurs technologies. Un exemple d'intégration WLAN/UMTS est montré afin de mieux voir les techniques de fusion entre deux réseaux.
- **Conception et analyse théorique des protocoles de handover** : avec notre structure hiérarchique, on a distingué deux types de handovers : un handover horizontal (*horizontal handover*) et un autre vertical (*vertical handover*). Un handover horizontal est un handover classique entre deux cellules homogènes de même niveau hiérarchique, tandis qu'un handover vertical se passe entre deux cellules hétérogènes. Dans la littérature, les protocoles pour les handovers horizontaux classiques sont bien étudiés et optimisés. Pour le handover vertical, deux protocoles sont proposés. La minimisation de temps de latence, la perte des paquets et l'overhead sont au cœur de notre conception.
- **Optimisation des protocoles de handover proposés** : il existe des applications exigeantes en termes de bande passante, délai et taux de perte. Le processus d'un handover peut dégrader la QoS demandée. Nous avons développé des stratégies plus agressives afin de réduire le maximum possible les taux de perte ainsi le temps de la latence durant le handover vertical.

Cependant, l'overhead de ces techniques est important. En combinant quelques techniques d'optimisations, on a pu obtenir à un protocole de pertes nulles, temps de latence faible et un overhead acceptable.

- **Conception des protocoles pour la gestion de mobilité** : le but de ces protocoles est de gérer la micro et macro-mobilité. Le protocole mobile IP est utilisé pour la macro-mobilité. Les protocoles de handover gèrent la micro-mobilité. Il reste le cas où la mobilité est rapide. Dans ce contexte, nous avons conçu un algorithme qui permet de prendre la bonne décision lors des handovers en se basant sur la vitesse des mobiles qui peut se traduire par le nombre des handovers effectués dans une période de temps.
- **Étude des performances** : un modèle de simulation simple mais efficace est développé sous OPNET. Plusieurs scénarios et paramètres de simulation sont considérés. Les paramètres utilisés reflètent exactement les caractéristiques de technologies réelles sans fil. Les résultats sont analysés en se basant sur une étude comparative par rapport aux analyses théoriques.

Pour la QoS dans les réseaux ad hoc, nos contributions sont originales et se situent au niveau de la couche IP. La démarche suivie est :

- **Conception et analyse des algorithmes pour le routage à multiple métriques** : le problème de trouver un chemin optimal sous la présence de deux ou plusieurs métriques additives et/ou multiplicatives est un problème NP-complet [5]. Il n'existe pas d'algorithme polynomial qui trouve toujours la solution optimale. Nous avons proposé une heuristique quasi-polynomiale basée sur l'algorithme de la relaxation lagrangienne. Cette heuristique s'est avérée théoriquement très intéressante en termes de temps d'exécution et qualité des chemins trouvés. Elle peut résoudre des problèmes à deux, trois et quatre métriques avec une garantie de passage à l'échelle. Par simulation, nous avons montré l'efficacité de cette méthode sous plusieurs scénarios. Cet algorithme est adopté par la suite dans notre protocole de routage avec QoS.
- **Conception et analyse d'un protocole de routage avec QoS<sup>1</sup>** : nous avons considéré le protocole OLSR comme le protocole de base pour ajouter une extension de QoS. Cette extension est nommée le protocole QOLSR. Chaque nœud mesure les métriques de QoS (bande passante, délai, etc) sur les liens vers ses voisins directs. Ces informations sont échangées localement afin de calculer un ensemble optimisé de nœuds relais. Cet ensemble permet par la suite à chaque nœud d'acquérir la topologie du réseau sous forme d'un graphe pondéré partiel et de calculer des routes optimales selon la demande des flux. Nous avons démontré que le protocole QOLSR permet de trouver des chemins optimaux dans une topologie partielle de réseau qui ont les mêmes performances que ceux trouvés en utilisant la topologie entière. Ceci est dû à notre algorithme de sélection des nœuds relais.
- **Étude des performances du protocole QOLSR** : un modèle de simulation dédié au protocole QOLSR est développé. Il permet de mesurer les performances de chaque technique utilisée dans ce protocole : le modèle d'interférence, le graphe de conflit, l'heuristique de sélection des nœuds relais, le fonctionnement entier du protocole. Nous avons mis en place plusieurs scénarios et paramètres de simulation. Plusieurs résultats ont été obtenus et sont analysés.

---

<sup>1</sup>Ce travail a été effectué en collaboration avec Anelise Munaretto.

- **Conception et analyse des méthodes de mesure des métriques de QoS** : des méthodes analytiques et d'autres expérimentales pour estimer la bande passante et le délai sur les liens dans un réseau ad hoc où chaque nœud est équipé par une carte sans fil 802.11 sont mises en place. Ces méthodes prennent en considération les interférences dans le voisinage (non seulement direct), l'algorithme exponentiel de Backoff, la taille des Buffers MAC, etc. Dans les méthodes analytiques, nous avons supposé que la couche MAC peut fournir des informations concernant l'état du canal. En revanche, les méthodes expérimentales<sup>2</sup> sont basées sur la théorie des graphes et aucune information n'est fournie par la couche MAC.
- **Conception et analyse d'un mécanisme de contrôle d'admission et de réservation** : le contrôle d'admission permet de vérifier la disponibilité des ressources avant d'accepter un flux QoS. Un protocole de signalisation est proposé pour contrôler, réserver et libérer les ressources d'une manière efficace. Les congestions, les interférences et la mobilité sont prises en considération.

Cette thèse constitue un travail de trois ans qui a été réalisé au sein du LRI et INRIA. Certains des travaux présentés ici sont encore en cours, d'autres ont déjà fait l'objet de publications dans plusieurs journaux et conférences internationales ou nationales. Un draft sur le protocole QOLSR est disponible dans le groupe MANET. Le modèle de QoS pour le réseau ad hoc que nous avons proposé est réalisé dans le cadre du projet SAFARI [6]. La contribution majeure du projet SAFARI est de proposer la continuité des services dans le monde ad hoc au travers d'une approche globale des besoins (d'IPv6 aux services applicatifs extensibles en passant par la QoS et la supervision). Cette contribution est le résultat des contributions dans chacune des briques qui constituent la plate-forme de services visée à savoir : les services d'interconnexion de base (IPv6, multicast sur la totalité du réseau), les services avancés (QoS dans la partie ad hoc, sécurité ad hoc et multicast), les services d'adaptation de contenus multimédia et de découverte dynamique des ressources et services.

---

<sup>2</sup>Ce travail a été effectué avec la collaboration d'Ignacy Gawedzki.



## Chapitre 2

# La mobilité dans les réseaux sans fil : état de l'art

Le GSM devrait céder sa place aux services GPRS et l'UMTS qui offrent des débits plus élevés et d'autres modes de communications en plus de la voix. Comme indiqué dans son nom, l'UMTS se veut universel. Il faut cependant se rendre à l'évidence : principalement à cause de sa bande passante assez limitée, cette 3ème génération ne sera pas la dernière, loin de là. L'usage de la solution WLAN nécessite une utilisation d'une bande de fréquence jusque là consacrée à la défense nationale, la migration de celle-ci vers de nouvelles bandes de fréquence est lente, ce qui empêche le WLAN de se développer librement. Dans ce contexte et afin de mieux positionner cette thèse, nous faisons une classification des réseaux mobiles et sans fil selon l'étendue géographique et la topologie. Cette classification va nous permettre d'introduire la notion d'une structure hiérarchique des réseaux mobiles et sans fil dans le chapitre suivant. Nous nous intéressons à l'aspect de l'évolution des systèmes sans fil et leurs caractéristiques pour mieux comprendre le besoin d'une nouvelle génération.

### 2.1 Historique, évolution et applications des réseaux sans fil

L'histoire des communications radio-mobiles peut être découpée en trois grandes phases. La première concerne les découvertes théoriques et la mise en évidence de l'existence des ondes radios. Cette phase débute en 1678 avec les travaux de Huygens sur les phénomènes de réflexion et de réfraction de la lumière. C'est Fresnel qui, en 1819, en démontre la nature ondulatoire. En 1865, Maxwell établit les célèbres formules unifiant phénomènes électrique, magnétiques et lumineux mais ce n'est qu'en 1887 que Hertz met en évidence pour la première fois la possibilité d'émission/détection d'une onde électromagnétique entre deux points distants de quelques mètres. À la fin du siècle, en 1897, Ducretet étend cette distance en établissant une liaison radio de quelques kilomètres de portée. Après avoir montré la possibilité de communications radio-mobile entre un bateau et une île en 1898, Marconi met en place la première radio trans-Atlantique entre l'Europe et les Etats-Unis trois ans plus tard. Il marque ainsi le point de départ des premiers systèmes de communications radio.

La seconde phase est constituée par le développement et l'évolution des équipements et des techniques mais pour des usages encore réservés à certaines catégories de la population. Les stations radio d'émission/réception sont au début du 20<sup>e</sup> siècle de taille imposante. Par exemple en 1902, une station radio militaire pour le télégraphe était constituée d'un moteur à essence pour

tracer le système de communications consistant en générateur de 1kW monté sur une remorque suivie d'une seconde remorque pour l'émetteur et le récepteur. C'est l'évolution des techniques et des équipements (taille, poids, la portée des communications et les services radiotéléphonie) qui permettra aux systèmes radios d'acquérir la dimension mobile. La seconde guerre mondiale va accélérer le développement des systèmes qui vont, dans les années 1950, se multiplier pour les applications civiles (compagnies de taxis et ambulances par exemple). Les équipements restent cependant encore lourds et occupent une place importante puisqu'ils sont généralement installés dans les coffres des véhicules.

Les progrès techniques et développement des systèmes de communications vont faire entrer les systèmes de communications sans fil et mobiles dans le domaine grand public (la troisième phase). Les premiers types de systèmes de communications disponibles au plus grand nombre sont les systèmes cellulaires. Conçu comme réponse à l'augmentation de la demande et à la faible disponibilité du spectre radio, les systèmes cellulaires analogiques se développent dans la décennie 1970. En 1979, le premier système cellulaire AMPS (*Advanced Mobile Phone Service*) est installé à Chicago, suivi en 1980 par le HCMTS (*High Capacity Mobile Telephone System*) à Tokyo. La décennie 1980 va voir ainsi se généraliser l'implantation de systèmes cellulaires analogiques dans de nombreux pays. Parallèlement, les systèmes sans cordon, se développent et connaissent des taux de croissance impressionnants. Toutefois, c'est réellement le système cellulaire numérique GSM (voir section 2.3.1), avec les services de type RNIS (*Réseau Numérique à Intergration de Services*) et la possibilité de roaming international, qui constitue le représentant le plus significatif de la révolution des mobiles de la décennie 1990. Les systèmes sans fil et mobiles vont progressivement tous les domaines d'activités économiques et intégrer peu à peu tous les types de services de télécommunications.

Comme on l'a déjà mentionné, plusieurs facteurs ont contribué à l'évolution rapide de ces dernières années. En effet, la miniaturisation des composants a permis de réduire la taille des produits électroniques sans fil en général (GPS, téléphone sans fil, ...) et le matériel informatique en particulier. Les ordinateurs portables sont de plus en plus légers et puissants en même temps, génération après génération. Cette miniaturisation s'est accompagnée avec une évolution d'autonomie et une baisse de consommation. Un autre facteur important qui aide beaucoup à rendre la technologie sans fil plus populaire est la baisse des prix ainsi que la disponibilité des bandes radios ISM (*Industrial Scientific and Medical*) et UNII (*Unlicensed National Information Infrastructure*) à 800 MHz, 2.4 GHz et 5 GHz, qui peuvent être utilisées gratuitement et sans autorisation (malgré quelques différences dans la législation d'un pays à un autre).

Avec la récente évolution des technologies de communication sans fil. On s'attend à un développement rapide des réseaux locaux sans fil et leur utilisation sera de plus en plus fréquente. Bien que les projets aient souvent débuté dans un cadre militaire pur, le domaine d'application des réseaux sans fil s'étend bien au-delà. En effet, les réseaux sans fil offrent une grande flexibilité ainsi qu'une rapidité et facilité de mise en place. Ils seront d'un grand apport lors des catastrophes naturelles, des incendies, où il sera indispensable de disposer rapidement d'un réseau pour organiser les secours et les opérations de sauvetage.

Les réseaux sans fil sont plus faciles à implanter dans des bâtiments, où il est impossible d'installer des câbles convenablement ; tel que les vieux bâtiments, les sites classés (exemple : châteaux et monuments historiques), lors des manifestations temporaires (congrès, foires, salons, expositions, le tour de France, et autre manifestation sportives). On peut aussi avoir recours à ce type de réseau lorsqu'on veut interconnecter des bâtiments à moindre coût (i.e., sans location d'une liaison spéciale chez un opérateur). On peut imaginer une application industrielle, où les nœuds seront en fait des robots mobiles qui pourront se déplacer librement dans l'usine. Ou bien encore,

dans des environnements hostiles à l'homme tels que des cratères de volcans pour surveiller leur activités ou bien le long d'une faille géologique. Dans les campus universitaires, l'utilisation des réseaux sans fil peut être très utile pour les étudiants qui pourront se connecter sur leurs comptes et travailler à partir de la bibliothèque ou leurs chambres.

## 2.2 Problèmes des transmissions radios et des réseaux sans fil

Les ondes radios (notées *RF* pour *Radio Frequency*) se propagent en ligne droite dans plusieurs directions. La vitesse de propagation des ondes dans le vide est de  $3 \cdot 10^8$  m/s. Lorsqu'une onde radio rencontre un obstacle, une partie de son énergie est absorbée et transformée en énergie (thermique par exemple), une partie continue à se propager de façon atténuée et une dernière peut éventuellement être réfléchi. L'atténuation augmente avec l'augmentation de la fréquence ou de la distance. De plus lors de la collision avec un obstacle, la valeur de l'atténuation dépend fortement du matériel composant l'obstacle. Généralement les obstacles métalliques provoquent une forte réflexion, tandis que l'eau absorbe le signal.

Les réseaux sans fil utilisent des ondes radios ou infrarouges afin de transmettre des données. Les transmissions radios dans les réseaux sans fil sont toutefois soumises à de nombreuses contraintes, liées à la nature de la propagation des ondes radios et aux méthodes de transmissions, rendant ce type de transmission non suffisante. Le signal transmis est sujet à nombreux phénomènes dont la plupart ont un effet de dégradation sur la qualité du signal. Cette dégradation se traduit en pratique par des erreurs dans les messages reçus qui entraînent des pertes d'informations pour l'utilisateur ou le système. Ces contraintes sont notamment :

- ☞ débit plus faible que celui du filaire. La bande passante est une ressource rare, il faut minimiser la portion utilisée pour la gestion du réseau, afin de pouvoir laisser le maximum de bande passante pour les communications ;
- ☞ les brouillages dus aux interférences : les liens radios ne sont pas isolés, deux transmissions simultanées sur une même fréquence ou, utilisant des fréquences proches peuvent interférer. De plus, les interférences peuvent venir d'autres types de machine non dédiées aux télécommunications. Par exemple, les fréquences utilisées dans les fours à micro-ondes sont dans les fréquences de la bande ISM ;
- ☞ les brouillages dus au bruit ambiant (que nous distinguons des interférences), provenant d'émission d'autres systèmes par exemple ;
- ☞ les évanouissements (ou *fadings*) dans la puissance du signal dus aux nombreux effets induits par le phénomène de multitrajets ;
- ☞ erreurs de transmission : les erreurs de transmission radio sont plus fréquentes que dans les réseaux filaires ;
- ☞ liens versatiles : les transmissions radios sont très sensibles aux conditions de propagation, ce que les rend versatiles. Un contrôle de la qualité des liens est obligatoire afin de pouvoir les exploiter convenablement pour les communications radios ;
- ☞ puissance du signal : la puissance du signal diminue avec la distance, et la puissance utilisée est sévèrement réglementée par les autorités compétentes des pays ;
- ☞ les pertes de propagation dues à la distance parcourue par l'onde radio, ou affaiblissement de parcours (*pathloss*) ;



- ☞ les atténuations de puissance du signal dues aux effets de masques (shadowing) provoqués par les obstacles rencontrés par le signal sur le trajet parcouru entre l'émetteur et le récepteur ;
- ☞ nœuds cachés : ce phénomène est très particulier à l'environnement sans fil. Les nœuds qui ne s'entendent pas, à cause d'un obstacle qui empêche la propagation des ondes, peuvent provoquer des collisions ;
- ☞ impossibilité de détecter les collisions au cours d'une transmission. Pour détecter une collision, une station doit être capable de transmettre et d'écouter en même temps. Or, dans les systèmes radios, il ne peut y avoir transmission et écoute simultanées ;
- ☞ consommation d'énergie : qui dit sans fil, dit aussi mobilité, et donc autonomie. Pour maximiser la durée de vie des batteries, il faut économiser autant que possible les transmissions inutiles ;
- ☞ sécurité : les détecteurs des signaux et les récepteurs passifs peuvent espionner les communications radio si ces dernières ne sont pas protégées ;
- ☞ mobilité et topologie dynamique : la disparition ou l'apparition d'un nœud ou un lien entre deux nœuds peut être le résultat d'un déplacement, c'est aussi parce que la batterie est épuisée ou lorsqu'une panne survient.

## 2.3 Les réseaux mobiles et sans fil

Les termes mobile et sans fil sont souvent utilisés pour décrire les systèmes existants, tels que le GSM, IS95, IEEE 802.11, Bluetooth, etc. Il est cependant important de distinguer les deux catégories de réseaux que recoupent les concepts de mobile et de sans fil de façon à éviter toute confusion. Prenez l'exemple (tiré de [7]) du téléphone sans cordon de résidence. Ce téléphone donne accès au RTC (réseau téléphonique commuté), le réseau classique de téléphone, ou au RNIS (Réseau numérique à intégration de services). Le support de communication utilise l'interface radio pour qu'un abonné puisse appeler depuis son jardin ou sa cuisine, mais ce dernier doit toujours rester au voisinage de son réseau d'abonnement. En cas de mobilité dépassant ces limites, l'utilisateur est contraint de contacter un opérateur local pour souscrire un nouvel abonnement ce genre de systèmes offrent le sans fil sans la mobilité.

Un utilisateur mobile est défini théoriquement comme un utilisateur capable de communiquer à l'extérieur de son réseau d'abonnement tout en conservant une même adresse. Certains systèmes tels que le GSM, IS95, UMTS offrent la mobilité et sans fil simultanément.

Un autre type de mobilité est celui mis en œuvre dans un réseau IP Mobile [8,9]. Le protocole IP Mobile est une extension d'IP permettant à un nœud de rester accessible avec la même adresse, tout en étant mobile dans différents réseaux. C'est-à-dire, le nœud mobile maintient son adresse permanente de réseau d'origine (*Home Address*) pour l'identification des connexions de la couche transport (par exemple *TCP*), et utilise une adresse temporaire (*Care-of-Address*) de réseau visité qui permet de router les paquets jusqu'à lui à son point d'attachement actuel sur Internet. L'adresse temporaire change à chaque fois que le mobile change de réseau d'attachement.

Il est possible de classer les réseaux sans fil de plusieurs manières, selon que nous nous intéressons à un critère ou à un autre. La figure suivante présente une classification possible selon les critères suivants : l'étendue géographique, la topologie et l'infrastructure utilisée.

Le concept cellulaire consiste à diviser une zone de couverture relativement grande, généralement appelée réseau sans fil large (WWAN, pour *Wireless Wide Area Network*) en plusieurs zones

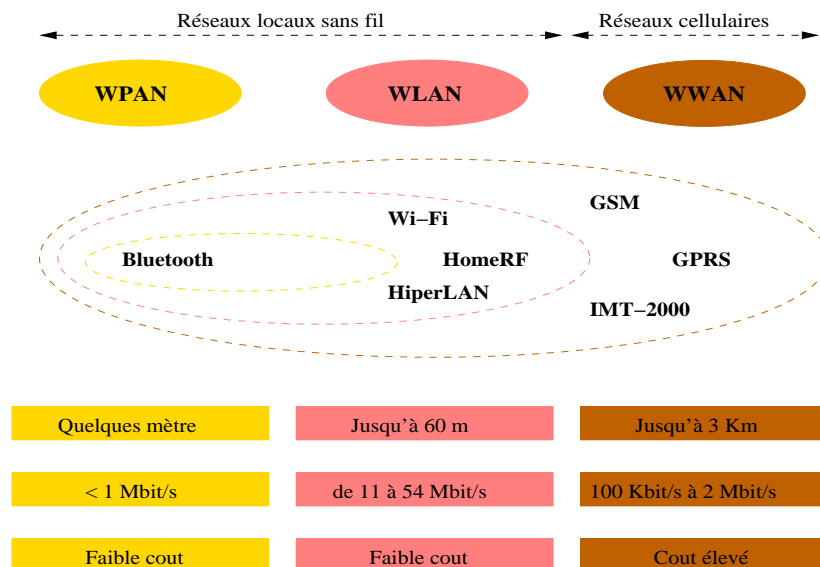


FIG. 2.1 – Les réseaux sans fil.

plus petites appelées cellules. Chaque cellule possède sa propre station de base (ou antenne relais). Chaque station de base utilise une bande de fréquence spécifique et gère l'ensemble des nœuds se trouvant dans la cellule. Les systèmes de communication cellulaires se sont développés en plusieurs générations. La première génération était analogique. La deuxième génération (2G) est numérique mais orienté mode circuit avec le standard le plus répandu dans le monde GSM (Global System for Mobile communication) qui offre un service de voix de 9,6kbits/s et qui permet un transfert de données à faible vitesse. Il existe une génération dite deux et demi (2G+) avec le GPRS (*General Packet Radio Service*) qui est une évolution du système GSM qui permet en agrégeant plusieurs canaux de voix d'offrir une commutation en mode paquet allant jusqu'à 115kbits/s. La troisième génération (3G) permet d'utiliser la commutation en mode paquet dans contexte multi-média. Le nom générique pour les différentes normes 3G est IMT-200 qui regroupe : (i) l'UMTS (Universal Mobile Telecommunications System) en Europe qui promet un transfert des données avec un débit théorique allant jusqu'à 2 Mbps dans les configurations à courte portée (picocellules) mais généralement 100kbit/s avec des cellules plus grande (micro-cellules), (ii) CDMA2000 aux Etats-Unis qui est une évolution de CDMAOne (*Code Division Multiple Access*) et W-CDMA (*WideBand CDMA*) au Japon, et (iii) EDGE (Enhanced Data rates for Global Evolution) parfois appelé E-GPRS (*Enhanced GPRS*) qui est une évolution du GPRS et du TDMA. Les propositions de la troisième génération ont été normalisées par les 3GPP (*3rd Generation Partnership Project*) et 3GPP2 [10].

Le concept du réseau local sans fil est né du besoin d'offrir par le sans fil les services habituellement accessibles par voie filaire (*Ethernet*). Au sein des entreprises ou des laboratoires, ces réseaux sont souvent utilisés en complément des réseaux filaires existants. Du réseau personnel (WPAN, pour *wireless Personnel Area Network*) au réseau local (WLAN, *Wireless LAN*), plusieurs standards existent : Bluetooth, IEEE 802.11, HiperLAN. Ces standards permettent aux mobiles de communiquer entre eux dans un même réseau d'accès, mais ils ne sont pas prévus généralement pour permettre de passer d'un réseau d'accès à un autre comme les systèmes mobiles cellulaires. Ils sont donc plus adaptés à des réseaux locaux dans des lieux identifiés tels qu'une salle de confé-

rence, un campus universitaire et des lieux de passage : aéroports, gares, etc.

### 2.3.1 Les réseaux cellulaires

La propriété d'atténuation, caractéristique de l'interface radio, a permis de développer le concept *cellulaire*. Dans ce modèle, la zone de couverture est divisée en cellules, à chaque cellule est affectée à une bande de fréquence. Du fait de la rareté du spectre hertzien, cette bande de fréquence est étroite, d'où la faible capacité de l'ensemble du système.

Pour faire face à l'augmentation ininterrompue du nombre d'utilisateurs des réseaux cellulaires, il a fallu tout à la fois accroître la capacité du système, diminuer la dimension des cellules et installer un nombre plus important de relais. Le GSM est le premier système cellulaire de téléphonie mobile efficace, économique et universel. Il a été très rapidement adopté par les utilisateurs ces dix dernières années. Le GSM évolue constamment vers le transfert de données (GPRS).

#### Le GSM

Le GSM (*Global System for Mobile communication*) [11] est un standard Européen établie par l'ETSI (European Telecommunications Standards Institute), qui est conçu essentiellement pour la téléphonie mobile sans fil, mais il peut supporter des transferts de données à faible débit (14.4 Kbits/s de base). Les mobiles ne communiquent pas directement entre eux mais ils doivent passer par la station de base, même s'ils sont proches. Le réseau global est formé d'une multitude de stations de base, chacune couvrant une cellule indépendante. Les cellules adjacente doivent utiliser des fréquences différentes pour éviter les interférences entre les communications.

Le GSM utilise la méthode d'accès F-TDMA (*Frequency Time Division Multiple Access*). Chaque mobile doit réserver un slot dans une première phase de contrôle avant de pouvoir commencer la communication. Le GSM utilise deux plages de fréquences différentes, la première est réservée pour les communications des mobiles vers les stations de base (890-915 MHz et 1710-1785 MHz pour le GSM 1800) et la deuxième dans le sens des stations de base vers les mobiles (935-960 MHz et 1805-1880 MHz pour le GSM 1800).

Certains choix technique du GSM sont fait en conséquence, notamment en matière d'architecture réseau et de mise en forme des ondes (modulation, codage, etc). Ces choix se révèlent toutefois contraignantes pour les services de données (transfert de fichier, vidéo, etc). Pour pallier ces limitations, le standard du GSM évolue sans cesse. Dans un premier temps, le GSM a standardisé des règles pour réaliser un transfert de données en utilisant les circuits de voix. Avec le HSCSD (High Speed Circuit Switched Data), on assiste à un premier développement du standard vers des débits supérieurs, mais toujours en mode circuit.

Pour améliorer encore l'efficacité de transfert de données, une évolution majeure du GSM est normalisée sous le nom de GPRS (*General Packet Radio Service*). Fondée sur l'interface radio du GSM, mais développant un partage de ressources dynamique adapté au trafic sporadique, le GPRS introduit une architecture réseau en mode paquet. Si l'utilisateur téléphone, l'information transit par le réseau cœur de type circuit téléphonique. Si l'utilisateur émet des paquets, ces derniers sont acheminés par le réseau cœur de type paquet. Enfin, EDGE (*Enhanced Data for GSM Evolution*) propose des débits supérieurs par l'introduction d'une modulation plus efficace, applicable à la fois au HSCSD et au GPRS.

L'association du GPRS et d'EDGE est souvent considérée comme un système 2.5, intermédiaire entre les systèmes 2G (GSM, etc) et 3G (UMTS, CDMA2000, etc).

## L'UMTS

L'apparition d'une troisième génération de réseaux de mobiles a pour origine tout à la fois la saturation des systèmes actuels et le besoin d'une couverture universelle ainsi que de services évolués tendant vers ceux offerts par les infrastructures fixes. Il faudrait ajouter à ces raisons une certaine volonté émanant de pays n'ayant pas forcément d'infrastructure fixe adéquate ou ayant raté le train du GSM d'entrer rapidement dans la troisième génération.

L'UMTS (*Universal Mobile Telecommunication System*) [12] est un système cellulaire de troisième génération qui fait partie de la famille IMT-2000 (International Mobile Telecommunication system 2000) et dont les spécifications techniques sont développées au sein du 3GPPP. L'architecture de ce système est composé essentiellement d'un réseau terrestre d'accès radio, l'UTRAN (*Universal Terrestrial Radio Access Network*) et d'un réseau cœur dérivé de celui spécifié pour la phase 2G+ du GSM. L'UTRAN utilise deux modes d'accès fondés sur la technologie CDMA large bande : l'UTRA/FDD (*Universal Terrestrial Radio Access/Frequency Duplex Division*) et l'UTRA/TDD (*Universal Terrestrial Radio Access/Time Duplex Division*).

L'UMTS est le résultat de plusieurs années de recherche qui ont été menées en Europe à partir de la fin des années 1980 et qui se poursuivent aujourd'hui à l'échelle mondiale. RACE (*Research of Advanced Communication Technologies in Europe*) a été le premier programme européen de recherche qui avait pour objectif de définir les principes techniques de l'UMTS [13]. Le Programme ACTS (*Advanced Communication Technologies and Services*) est venu prendre le relais à partir de 1995.

L'UMTS est conçu pour pouvoir s'interconnecter et interagir avec le GSM dans toutes ses phases d'évolution. Aussi, l'UMTS peut être considéré comme l'étape la plus porteuse en terme d'innovations technologiques. Ce système se distingue en effet de ses prédécesseurs HSCSD, GPRS et EDGE, par des débits plus élevés, une couverture plus large, une capacité en termes de nombre d'abonnés plus grande et des services plus variés.

### 2.3.2 IP Mobile

IP Mobile [8, 9] est basé sur la distinction de l'identification du mobile et son attachement physique à un réseau d'accès IP. Lorsqu'un utilisateur change de sous-réseau, il est nécessaire de modifier son préfixe pour que les routeurs puissent acheminer l'information au nouveau sous-réseau. Dans cette optique, le mobile possède deux adresses : son adresse permanente liée à son réseau d'abonnement (*Home Address*) est utilisée pour identifier les connexions des couches les plus élevées (par exemple TCP, UDP), tandis qu'une adresse temporaire (*Care-of-address*) liée au réseau visité permet de router les paquets jusqu'à lui via son point d'attachement actuel sur Internet. L'adresse temporaire change chaque fois que le mobile change de réseau d'attachement. Le protocole IP Mobile définit deux types d'entités réseau chargées de gérer la mobilité :

- **Home Agent (HA)**. Routeur dans le réseau d'abonnement (*Home Network*) du mobile, qui met à jour une information de localisation du mobile et qui envoie par un tunnel les paquets destinés à ce mobile, lorsque ce dernier est en dehors de son réseau d'abonnement.
- **Foreign Agent (FA)**. Routeur situé dans le réseau visité (*Foreign Network*) par le mobile. Il décapsule et relaie les paquets destinés au mobile visiteur.

Le fonctionnement d'IP Mobile est caractérisé par trois étapes essentielles (cf. figure 2.2, tirée de [14]) :

- ☞ Obtention d'une adresse temporaire (*COA : Care-Of-Address*) pour les nœuds mobiles permettant leur localisation.

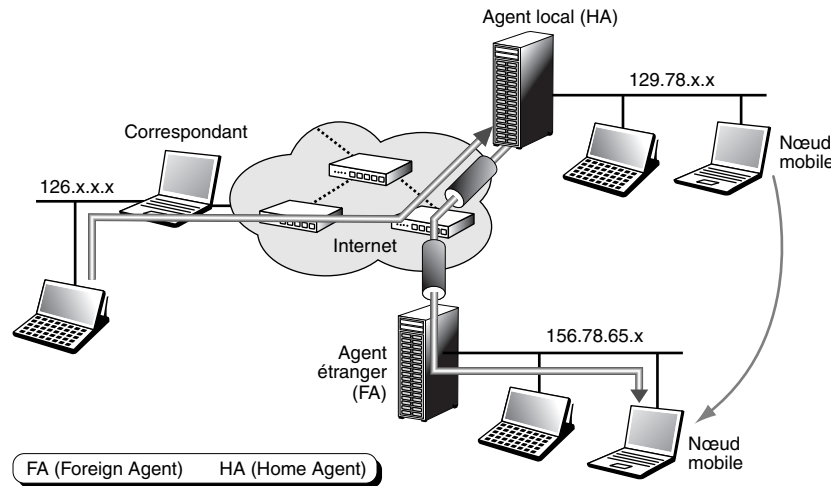


FIG. 2.2 – IP Mobile.

- ☞ Enregistrement de cette adresse auprès du Home Agent.
- ☞ Encapsulation des paquets arrivant au réseau d'abonnement en utilisant cette adresse temporaire.

À chaque changement de point d'accès (c'est-à-dire, *Handoff* au niveau de la couche liaison) le mobile détecte son déplacement dans le nouveau sous-réseau, reçoit une *COA* et doit s'enregistrer auprès de son *Home Agent*. Durant cette période, soit les paquets sont perdus parce que le précédent point d'accès n'a pas de route pour joindre le mobile ou celle-ci n'est plus valide, soit les paquets sont transmis par le point d'accès précédent au nouveau point d'accès comme proposé dans [15]. Ceci est appelé *smooth handoff* et suppose que le point d'accès précédent est averti du déplacement du mobile.

Le routage des paquets vers la nouvelle localisation de l'utilisateur est peu optimal du fait du passage des paquets par le réseau d'abonnement de l'utilisateur. Une optimisation de la route est proposée pour réduire la signalisation engendrée par IP Mobile dans le réseau Internet [16].

Dans un réseau micro-cellulaire (ex. *Wi-Fi*) et pico-cellulaire (ex. *Bluetooth*) où chaque point d'accès couvre une zone géographique relativement petite, il faudrait s'attendre à ce que les mobiles exécutent très souvent des *Handoffs* IP Mobile. Ceci dégraderait les performances du réseau et la continuité du service au niveau de ces mobiles. Pour pallier ces problèmes et rendre IP Mobile plus rentable, la gestion de la mobilité est divisée en macro-mobilité et micro-mobilité. La macro-mobilité désigne le déplacement inter-domaine, c'est-à-dire d'un domaine administratif IP à un autre (par exemple, entre réseaux d'accès sans fil étendus) par contre la micro-mobilité représente le déplacement intra-domaine, c'est-à-dire à l'intérieur d'un même domaine administratif IP (voir figure 2.3).

IP Mobile est bien adapté pour la gestion de la macro-mobilité, mais il est beaucoup moins pour la gestion de la micro-mobilité. De nombreuses propositions sont faites pour assurer la gestion de la micro-mobilité classées en trois catégories : (i) Protocoles basés sur une hiérarchie des agents [17]. Le domaine visité est structuré en une hiérarchie d'agents. Ces protocoles sont caractérisés par l'utilisation de l'encapsulation IP pour la communication entre les différentes parties de la hiérarchie. Le mobile utilise une care of address locale au domaine visité ; (ii) Protocoles basés sur un marquage de route. Ces protocoles utilisent un acheminement IP par marquage de route

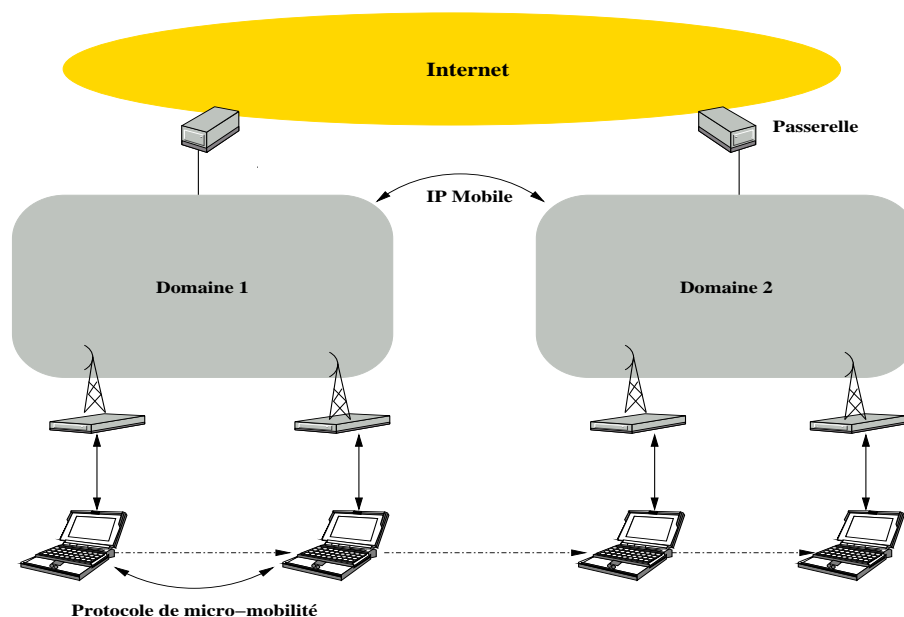


FIG. 2.3 – Macro-mobilité et Micro-mobilité.

dans le réseau d'accès. L'établissement de la route se fait par la transmission de proche en proche de paquets spécifiques d'un mobile vers la passerelle. Cette transmission permet aux nœuds intermédiaires de retenir le chemin de manière à l'utiliser en sens inverse pour joindre le mobile. En général le mobile garde sa *Care-of-Address* enregistrée auprès de son *Home Agent*. Parmi ces protocoles on trouve le IP cellulaire [18–20], HAWAII [21, 22] et TeleMIP [23] ; (iii) Protocoles basés sur les réseaux MANET (*Mobile Ad hoc NETwork*). Ces protocoles sont conçus pour les réseaux d'accès ad hoc, où les hôtes et les routeurs sont mobiles. Le routage est multi-sauts et s'adapte au fur et à mesure que le mobile se déplace et que la topologie change.

### 2.3.3 Les réseaux privés sans fil

Les réseaux locaux sans fil se sont des réseaux géographiquement limités, dans lesquels les terminaux se trouvent tous dans un milieu restreint en taille, correspondant soit à une entreprise, soit à un environnement personnel. Plusieurs possibilités peuvent être envisagées pour la réalisation de ces réseaux : soit la communication hertzienne s'effectue sur l'ensemble du site, soit les communications s'effectuent à l'intérieur de petites cellules reliées entre elles. Les communications entre les équipements terminaux peuvent s'effectuer directement ou par le biais d'une borne intermédiaire. Quant aux communications entre bornes de concentrations, elles peuvent s'effectuer de façon hertzienne ou par câble.

Les réseaux locaux sans fil connaissent actuellement d'important développement de fait de la flexibilité de leur interface, qui permet à un utilisateur de se déplacer dans son entreprise, tout en restant connecté. Plusieurs gammes de produits sont actuellement commercialisées, et plusieurs tendances se dégagent : les réseaux sans fil, qui desservent les équipements d'un seul utilisateur (les réseaux personnels, ou PAN : *Personal Area Network*), ceux qui desservent une entreprise et ceux qui connecte des utilisateurs sur une distance métropolitaine. Ces réseaux peuvent atteindre des débits de plusieurs mégabits par seconde, voire de plusieurs dizaines de mégabits par seconde.

La normalisation devrait avoir un fort impact sur les réseaux locaux sans fil. Aux États-Unis,

ce sont principalement deux groupes de travail de l'IEEE (*Institut of Electrical and Electronics Engineers*), l'IEEE 802.11 et l'IEEE 802.15, qui sont en charge de cette normalisation, tandis que le groupe HiperLAN (*High Performance Local Area Network*) s'en occupe en Europe.

## Bluetooth

Le standard Bluetooth [24] a été initié par Ericsson et un groupe de travail réunissant plusieurs grands industriels (dont 3Com, IBM, Nokia, Toshiba, etc). C'est une technologie peu onéreuse grâce à la forte intégration des composants électroniques. Les fréquences utilisées sont comprises entre 2400 et 2483,5 MHz. La même gamme de fréquences se retrouve dans la plupart des réseaux sans fil utilisés dans un environnement privé, que ce dernier soit personnel ou d'entreprise. Cette bande ne demande pas de licence d'exploitation.

Bluetooth est destiné à un usage qui remplacera les câbles de connexion entre les machines ou entre les machines et leurs accessoires (par exemple : connexion entre un ordinateur, le clavier, ou bien la souris, ou encore l'imprimante, etc ; ou encore communication entre la télévision et le lecteur DVD). Plus généralement Bluetooth permet de construire des petits réseaux personnels de quelques dizaines de mètres de portée.

Le réseau de base appelé *piconet*, est formé d'un maître et plusieurs esclaves (jusqu'à sept esclaves). C'est le maître qui se charge de gérer les communications comme dans le cas des réseaux à station de base. Plusieurs *piconets* peuvent être interconnectés pour former un scatternet, d'après le mot anglais *scatter*, dispersion. Dans ce scatternet, un esclave peut être l'esclave de plusieurs maîtres à la fois, en plus, un maître d'un *piconet* peut devenir un esclave du maître d'un autre *piconet*. La figure 2.4 tirée de [7], donne un exemple de connexion de terminaux Bluetooth.

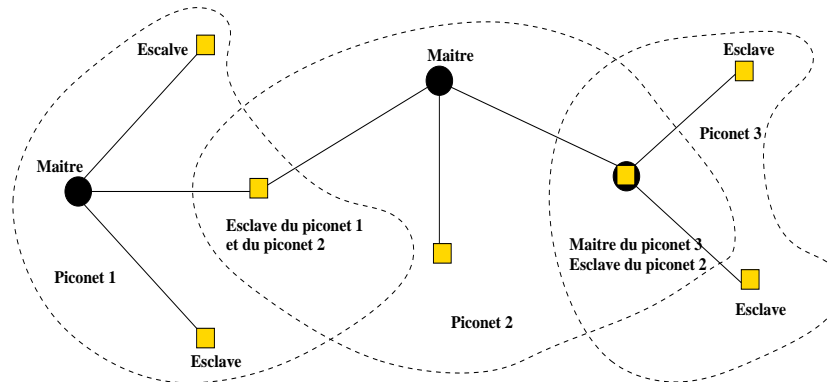


FIG. 2.4 – Schéma de connexion de terminaux Bluetooth.

Le débit à l'intérieur d'un *piconet* peut atteindre près de 1 Mbps entre les terminaux participant à la communication. Le temps est découpé en slots égaux. Un terminal utilise une fréquence et un slot puis, par un saut de fréquence (*Frequency Hop*), il change de fréquence sur la tranche de temps suivante, et ainsi de suite. C'est le maître qui sollicite ses esclaves pour parler d'une façon synchronisée.

La proposition de Bluetooth a été prise en compte par l'IEEE dans le groupe de travail IEEE 802.15 qui s'occupe de la standardisation des réseaux personnels (*PAN : Personal Area Network*). D'autres sous-groupes de travail de l'IEEE 802.15 se concentrent sur des applications particulières. Par exemple, Car Profile s'intéresse aux communications à l'intérieur d'une voiture, et le sous-

groupe AV (Audio/Video) à la problématique du transport de parole de bonne qualité ou de CD audio ainsi qu'à l'échange de vidéo.

Le Bluetooth souffre de quelques problèmes liés à la faiblesse de : débit, peu attrayant, la technique de partage de l'interface radio et en fin la concurrence très vive de la norme IEEE 802.11.

### L'IEEE 802.11

La norme 802.11 est le standard IEEE décrivant les caractéristiques d'un réseau local sans fil (WLAN) [25, 26]. Cette norme est connue sous l'appellation commerciale Wi-Fi qui signifie Wireless Fidelity. La norme 802.11 est le standard des couches inférieures à IP le plus utilisé dans les réseaux ad hoc (section 2.4). Dans cette section nous décrivons les différents modes de fonctionnement de cette norme. Dans un réseau IEEE 802.11, deux types d'architecture sont définis et correspondent à deux modes de fonctionnement :

- **Le mode infrastructure.** Dans ce mode, les mobiles communiquent directement avec un point d'accès fixe. Ces points d'accès sont analogues aux stations de bases des réseaux cellulaires et sont généralement reliés entre eux par un réseau filaire ou hertzien. Un point d'accès est le point de passage obligé pour qu'un mobile puisse communiquer avec un autre mobile. Les mobiles peuvent se déplacer tout en restant dans la zone de couverture du point d'accès.
- **Le mode ad hoc.** Dans ce mode, les mobiles communiquent directement entre eux sans avoir recours à une tierce station, à condition qu'ils soient à porté radio. Dans la figure 2.5, si un nœud A sort de la portée radio d'un autre nœud B, il ne pourra pas communiquer avec B.

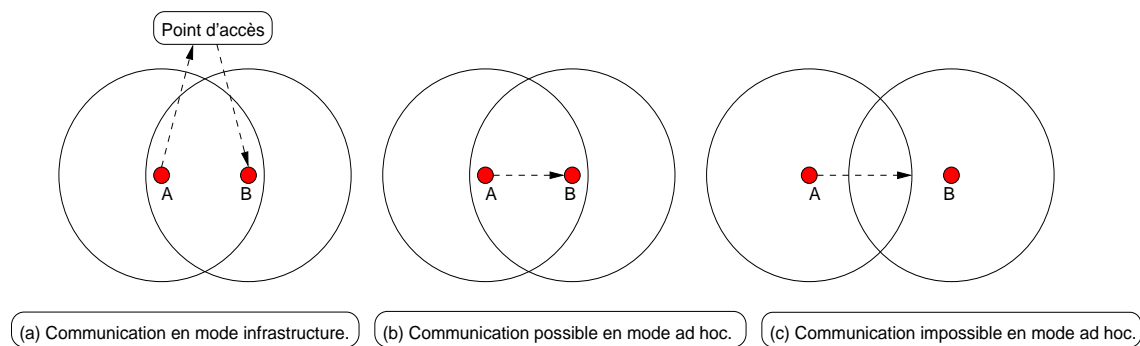


FIG. 2.5 – Exemple de modes d'opération du IEEE802.11.

Comme tous les standards de l'IEEE, l'IEEE 802.11 couvre les deux premières couches du modèle Open System Interconnection (OSI). Une des caractéristiques essentielle du standard est qu'il définit une couche Medium Access Control (MAC) commune à toutes les couches physiques. Ainsi de futures couches physiques pourront être ajoutées au standard sans que de nouveaux changements soient apportés à la couche MAC.

La couche MAC 802.11 d'un réseau ad hoc utilise la méthode d'accès au medium appelée DCF (*Distributed Coordination Function*). DCF est basée sur la technique CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) et offre d'autres fonctions qui sont confiées normalement aux couches supérieures, comme la fragmentation, les retransmissions et les accusés de réception. Dans un environnement sans fil, les collisions ne sont pas détectées du fait qu'un émetteur ne



peut pas transmettre et écouter en même temps. Pour éviter les collisions, le CSMA/CA utilise différents mécanismes, tels que l'écoute du support physique, l'algorithme de backoff pour gérer l'accès au canal, la réservation du canal et les trames d'acquiescement positif.

Dans l'IEEE 802.11, l'écoute du support se fait à la fois au niveau de la couche physique avec le PCS (*Physical Carrier Sens*) et au niveau de la couche MAC avec le VCS (*Virtual Carrier Sense*). Le PCS détecte la présence d'autres stations 802.11 en analysant toutes les trames passant sur le support hertzien et en détectant l'activité sur le support grâce à la puissance relative du signal des autres stations. Le VCS est un mécanisme de réservation basé sur l'envoi de trames RTS/CTS (*Request To Send/Clear To Send*) entre une station source et une station destination avant tout envoi de données.

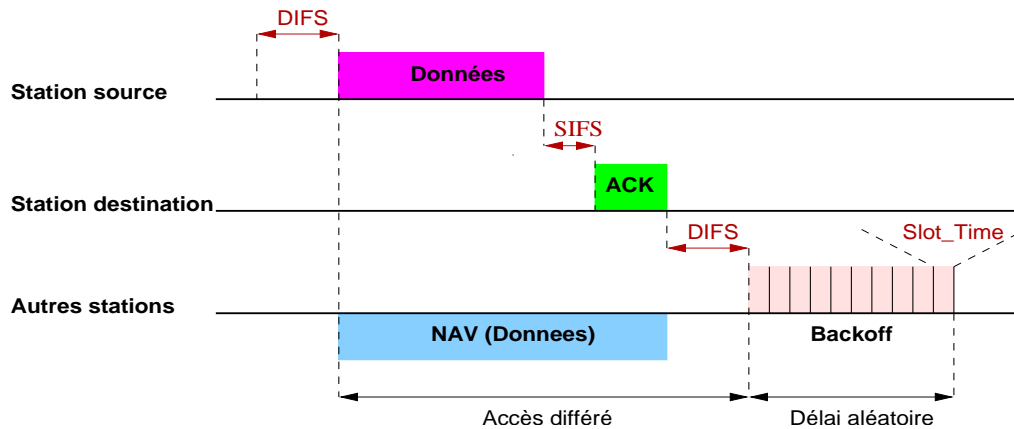


FIG. 2.6 – La méthode de base de transmission des données.

La méthode de base de transmission d'un paquet par une station 802.11 (*two-way handshake scheme*) n'utilise pas le mécanisme de réservation. Lorsqu'une station source veut transmettre des données à une station destination, elle vérifie si le support est libre. Si aucune activité n'est détectée pendant une période de temps correspondant à un DIFS (*Distributed InterFrame Space*) alors la station source transmet ses données immédiatement. Si le support est toujours occupé, la station continue de l'écouter pendant une période plus longue qu'un DIFS, jusqu'à ce qu'il soit à nouveau libre. Dès que le support est enfin libre pendant un temps correspondant à un DIFS, la station retarde encore sa transmission pendant un temps backoff généré aléatoirement puis transmet ces données vers la station destination. Si les données envoyées ont été reçues de manière intacte (la station destination vérifie le CRC de la trame de données), la station destination attend pendant un temps équivalent à un SIFS et émet un ACK pour confirmer la bonne réception des données. Si ce ACK n'est pas détecté par la station source ou si les données n'ont pas été reçues correctement ou encore si ce ACK n'a pas été reçu correctement, alors on suppose qu'une collision s'est produite et les données sont re-transmises après un certain temps aléatoire. Pendant toute la durée de la transmission entre la station source et la station destination, les autres stations mettent à jour leur NAV (Network Allocation Vector), en incluant le temps de transmission de la trame de données, le SIFS ainsi que le ACK. En plus, pour éviter la monopolisation du canal, La station 802.11 doit attendre un temps backoff aléatoire entre la transmission de deux nouveaux paquets (paquets ne font pas parti de la même session). La figure 2.6 montre le mécanisme de base *two-way handshake*.

Le VCS permet de réserver le support entre deux stations avant tout envoi de données (*four-way handshake scheme*). Lorsqu'une station source veut transmettre des données, elle envoie tout d'abord une trame de réservation RTS. Toutes les stations entendant cette trame, lisent le champ

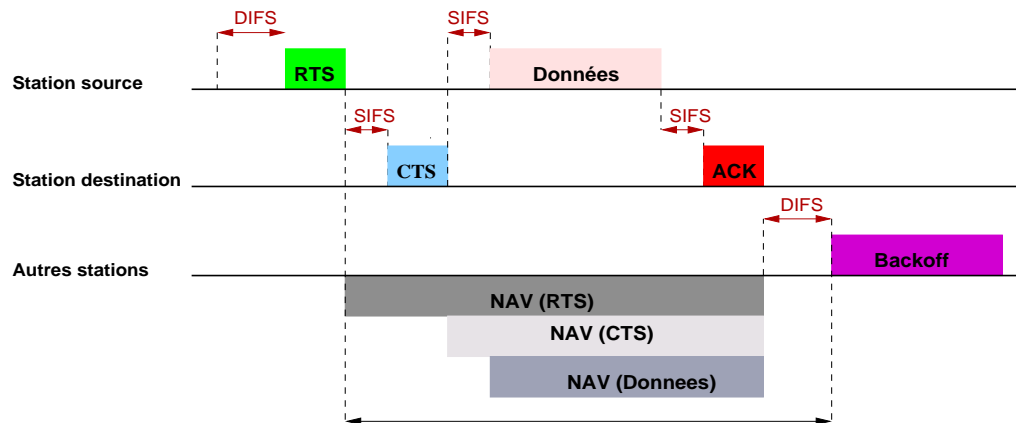


FIG. 2.7 – Transmission de données en utilisant les trames RTS/CTS.

de durée du RTS et mettent à jour leurs NAV. Dans la figure 2.7, lorsque la station destination reçoit le RTS, elle répond, après avoir attendu pendant un SIFS, en envoyant une trame CTS. Les autres stations entendant le CTS, lisent le champ de durée du CTS et mettent à nouveau à jour leur NAV. Après la réception du CTS par la station source, celle-ci est assurée que le support est stable et réservé pour sa transmission des données qui peut alors débiter.

Ce mécanisme permet ainsi à la station source de transmettre ces données ainsi que de recevoir le ACK sans qu'il n'y ait aucune collision. Comme les trames RTS/CTS réservent le support pour la transmission d'une station, elles sont utilisées habituellement lorsque l'on a de grosses trames à envoyer pour lesquelles une retransmission serait trop coûteuse en terme de bande passante. Les stations peuvent choisir d'utiliser le mécanisme RTS/CTS que lorsque la trame à envoyer excède une variable *RTS\_Threshold*.

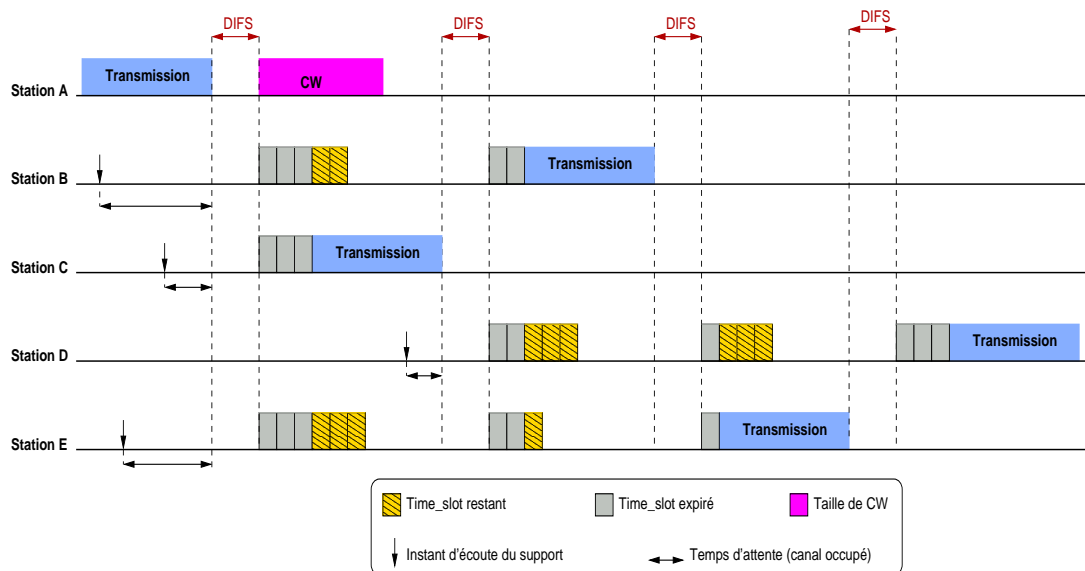


FIG. 2.8 – L'algorithme de backoff.

Dans 802.11, le temps est découpé en tranche de temps qui correspondent à un *Slot\_Time*. Il représente le temps nécessaire pour détecter la transmission d'un paquet par une autre station.

Donc, il dépend fortement de la couche physique, délai de propagation et le temps pour signaler l'état de canal à la couche MAC.

Sans le mécanisme RTS/CTS, lorsque deux stations accèdent en même temps au support, une collision se produit. Ces deux stations utilisent alors l'algorithme de Backoff pour savoir quand est ce qu'elles vont pouvoir à nouveau accéder au support. L'algorithme de Backoff définit une fenêtre de contention (CW : Contention Window) qui correspond au nombre de time-slots qui peuvent être sélectionnés pour le calcul du temps d'attente (appelé temporisateur de Backoff). Lorsque le support est libre, les stations décrémentent leurs timers jusqu'à ce que le support soit occupé ou jusqu'à ce que le timer atteigne la valeur 0. Si le support devient libre pendant une période supérieure à un DIFS après le blocage de la décrémentation (i.e., le support a été occupé), les stations peuvent à nouveau faire décrémenter leurs Backoff timer. Pour chaque transmission d'un paquet, le Backoff timer est sélectionnée uniformément entre  $[0, CW-1]$ . La valeur de CW dépend du nombre d'échecs de transmission d'un paquet, c'est-à-dire, pour chaque paquet stocké pour une transmission, la fenêtre CW prend une valeur initial  $CW_{min}$  et elle sera doublé à chaque échec de transmission, jusqu'à une valeur maximal  $CW_{max}$  (les valeurs de  $CW_{min}$  et  $CW_{max}$  sont spécifiées par la couche physique). La valeurs de CW conserve  $CW_{max}$  pour le reste des tentatives. Cet algorithme est appelé le schéma exponentiel du Backoff. Grâce à cet algorithme, les stations ont la même probabilité d'accéder au support car chaque station doit y accéder à nouveau après chaque transmission. Le mécanisme de Backoff est illustré par la figure 2.8.

## HiperLAN

HiperLAN (*High performance radio Local Area Network*) est un standard ETSI [27], pour les réseaux locaux. La famille HiperLAN comprend quatre propositions, notamment HiperLAN1, utilisée à l'intérieur des bâtiments sur des distances de l'ordre de 50m. Le déplacement des utilisateurs ne doit dépasser 10m/s. HiperLAN2 étend la distance à 200m, et le débit passe à 23.5Mbps. Le rôle de cette génération est d'accéder aux réseaux ATM. HiperLAN3 permet le point à multi-point. La distance entre stations peut atteindre 5km avec un débit supérieur à 20Mbps par utilisateur. Enfin, HiperLAN4 veut remplacer les liaisons ATM sur des distance de 150 à 200m, avec un débit compatible avec les normes de l'ATM, c'est-à-dire 155Mbps. Il est à noter qu'il n'existe pas de produit commercial conforme à cette norme.

## 2.4 Résumé

Les systèmes de communication cellulaires se sont développés en plusieurs générations. Le GSM est conçu pour la téléphonie vocale. C'est pourquoi il est fondé sur la commutation de circuits. Pour améliorer encore l'efficacité de transfert de données, une évolution majeure du GSM est normalisée sous le nom de GPRS. GPRS introduit une architecture réseau en mode paquet et permet des débits un peu plus élevés. Mais c'est la troisième génération de réseaux sans fil baptisée UMTS, qui devra réellement imposer la convergence de tous les types de communications avec des débits importants. Ces systèmes offrent la mobilité et sans fil simultanément.

Un autre type de mobilité est celui mis en œuvre dans un réseau IP Mobile. Il permet de router les paquets vers un mobile dans un réseau visiteur sans modifier d'adresse IP. Il est bien adapté pour la gestion de la macro-mobilité, mais il est beaucoup moins pour la gestion de la micro-mobilité.

Les réseaux locaux sans fil se sont des réseaux géographiquement limités. Ils connaissent actuellement d'important développement de fait de la flexibilité de leur interface, qui permet à un uti-

lisateur de se déplacer dans son entreprise, tout en restant connecté. Plusieurs standards existent : Bluetooth, IEEE 802.11, HiperLAN. Ces standards permettent aux mobiles de communiquer entre eux dans un même réseau d'accès.

## 2.5 Conclusion

Téléphoner avec un mobile GSM ou GPRS aujourd'hui n'apporte aucune différence, les coupures sous les ponts, dans les tunnels et les escaliers ne disparaissent pas et la qualité de la communication n'est ni pire ni meilleure. En revanche, la réception d'e-mails, la visite des sites Wap, la consultation d'information en tout genre et le téléchargement d'applications se trouvent considérablement optimisée par l'utilisation de GPRS. Après un engouement sans précédent en Europe pour le projet UMTS, la mise en œuvre des téléphones mobiles de 3G a commencé. Les réseaux 3G sont compatibles avec les réseaux GSM/GPRS. Ils permettant d'assurer la continuité de service (voix, SMS, MMS, data) dans les zones non couvertes par l'UMTS. Dans la pratique, un téléphone 3G passe en mode GSM dès qu'il se trouve hors d'une zone de couverture UMTS, et inversement. Il est compatible avec les réseaux GSM/GPRS. Le retard pris par le GPRS et l'UMTS a permis au WLAN de se développer rapidement et de présenter une technologie alternative. L'intégration des réseaux 2G/3G et les réseaux WLAN lance un réel challenge. Le prochain chapitre présentera un protocole de handover pour passer d'un réseau cellulaire à un réseau WLAN et inversement ou au sein de même réseau WLAN.

Parallèlement aux réseaux cellulaires, d'autres systèmes ont voulu offrir la mobilité et la transmission sans fil. D'après [7], Le monde de l'IETF a pris en compte le succès de la mobilité universelle du GSM et proposé un protocole IP Mobile. À l'instar du roaming de GSM, IP Mobile garantit à un utilisateur la possibilité de se connecter de n'importe où sur le réseau Internet sans pour autant imposer de changement d'adresse IP. D'autres courants dans l'IETF ont œuvré pour garantir une mobilité sans fil au sein d'une zone relativement limitée.



## Chapitre 3

# La 4G : l'interopérabilité entre les réseaux

Fondé sur un cœur de réseau IP, le futur système de télécommunication 4ème génération (4G) représente la convergence entre le réseau de 3ème génération (type UMTS) et les diverses technologies radio complémentaires. L'objectif est de fournir aux utilisateurs des services sans interruption dans un environnement hétérogène. Les utilisateurs mobiles peuvent accéder ou échanger des informations indépendamment de leur position, du temps et en utilisant des équipements différents en terme de capacité. Les réseaux 4G sont des réseaux hybrides qui intègrent différentes topologies et plate-formes réseaux.

Il existe deux types d'intégration : l'intégration des différents types de réseaux sans fil hétérogènes (wireless LAN, WAN, PAN ainsi que les réseaux ad hoc) ; et l'intégration des réseaux sans fil avec l'infrastructure fixe, l'Internet et le réseau téléphonique fixe. Cependant, beaucoup de travaux demeurent pour permettre une intégration sans couture.

La gestion de la mobilité est une opération fondamentale pour n'importe quel réseau mobile. Bien que les fonctionnalités et les implémentations varient entre les diverses technologies, certaines caractéristiques de base sont communes (voir section 3.1). Dans cette thèse, nous évoquons la nécessité d'une structure hiérarchique des réseaux sans fil hétérogènes, afin de mieux gérer la mobilité entre les réseaux hétérogènes. Cette structure est virtuelle et nécessite de nouvelles composantes et fonctionnalités pour gérer la communication entre les réseaux hétérogènes. Nous prenons comme exemple, l'interconnexion UMTS/WLAN afin de montrer les techniques réelles d'intégration. Ensuite, nous supposons que l'intégration est faite et nous nous focalisons sur les transferts cellulaires entre les différentes technologies des réseaux sans fil tout en tenant compte de leurs diversités (débit, latence, zone de couverture, etc).

### 3.1 Les principes de la gestion de mobilité dans les réseaux sans fil et mobiles

L'objectif principal de la gestion de mobilité est de maintenir des informations sur la position des terminaux mobiles et de gérer leurs connexions lorsqu'ils se déplacent dans les zones de couvertures. La gestion de la mobilité inclut deux procédures [28] : la gestion de la localisation et la gestion des handovers. La gestion de la localisation permet de fournir au réseau des informations sur la position courante d'un terminal mobile. Cette fonctionnalité comprend : i) *le processus d'inscription de la localisation*, où le terminal mobile est authentifié et sa position est

mise à jour, ii) *le paging*, où la position du terminal mobile est recherchée pendant l'initialisation d'une nouvelle session.

Le handover est le processus par lequel une communication établie est maintenue alors que le terminal mobile se déplace à travers le réseau cellulaire ; elle implique que la communication puisse passer d'un canal physique à un autre avec une coupure sans conséquences (en moyenne < 100 ms pour une communication voix dans le GSM).

Dans un réseau cellulaire, la station de base (BS : *Base Station*) constitue le point d'accès de toutes les communications. Elle remplit le rôle de serveur pour les clients se trouvant dans sa cellule. Lorsqu'un utilisateur mobile quitte sa cellule pour entrer dans une autre cellule, il peut être amené à s'approcher d'une BS voisine. Pour poursuivre sa communication, cet utilisateur se voit contraint de changer de point d'accès. Il effectue alors ce que l'on appelle un *transfert intercellulaire*, ou *handover* ou encore *handoff*, qui consiste à demander à un gestionnaire du réseau, ou commutateur, de mettre en place la signalisation nécessaire au transfert. Ce type de handover se passe entre deux stations de bases qui utilisent la même technologie sans fil.

Le processus du handover peut être divisé en trois étapes : *initiation*, *décision* et *exécution* [29]. La phase d'initiation du handover est déterminée par des conditions spécifiques comme la qualité du signal, la disponibilité d'un point d'accès alternatif, etc.

La décision de s'attacher à un nouveau point d'accès peut être prise de trois manières différentes : un handover contrôlé par le réseau, un handover assisté par le terminal mobile et un handover contrôlé par le terminal mobile [30, 31]. Dans les handovers contrôlés par le réseau, c'est le réseau qui décide selon les mesures effectuées sur les signaux radio envoyés par les terminaux mobiles. Puisque cette solution est complètement centralisée, elle nécessite une puissance considérable de calcul. Elle est aussi pénalisée par l'absence d'une connaissance sur les conditions récentes de chaque terminal.

Afin d'épargner le réseau de la complexité de la tâche, le handover assisté par le terminal mobile permet au terminal d'effectuer les mesures et au réseau de prendre les décisions.

Dans un handover contrôlé par le terminal mobile, le terminal possède l'autorité et l'intelligence pour décider selon ses propres mesures. Cette solution a l'avantage d'être distribuée mais elle a des impacts sur la stabilité et la sécurité du réseau.

L'exécution d'un handover exige un échange de signalisation pour rétablir la communication et ré-acheminer les paquets de données via le nouveau point d'attachement.

### 3.2 Une structure hiérarchique des réseaux sans fil

Les technologies actuelles des réseaux sans fils varient fortement selon la bande passante offerte, le temps de latence, la fréquence utilisée, les méthodes d'accès, la puissance de consommation, la mobilité, etc. En tenant compte de l'hétérogénéité, l'ensemble de ces technologies peut être divisé en deux catégories : (i) ceux qui offrent des services avec un faible débit mais sur une large zone géographique ; (ii) ceux qui fournissent des services avec un débit important en couvrant des petits secteurs. Aucune technologie sans fil ne peut garantir toute seule un débit adaptable pour chaque application, nombreux services de données avec une connexion sans coupures, un nombre important d'utilisateurs sur une large zone de déplacement et une latence minimale lors de changement des cellules. Un terminal mobile équipé d'une interface WLAN se trouve dans un milieu restreint en taille (entreprise ou environnement personnel), il ne peut alors pas répondre aux besoins de bande passante de certaines applications dans le cas où il possède une interface WWAN.

Une solution possible consiste à faire coexister et combiner des technologies différentes de

réseaux sans fil pour obtenir une meilleure couverture et si possible un choix dynamique de réseau selon le type d'application. Dans ce cas, le terminal mobile est doté de plusieurs interfaces lui permettant de se déplacer librement et d'accéder aux infrastructures des réseaux filaires à travers des réseaux sans fil alternatifs. Prenons l'exemple d'un utilisateur possédant un ordinateur portable (*laptop*) ou un PDA (Assistant numérique personnel) connecté à Internet dans son bureau via une technologie infrarouge (permet de relier deux dispositifs dotés de ports à infrarouge). Lorsqu'il quitte son bureau avec sa machine, il se trouve dans le couloir de son entreprise avec une connexion Wi-Fi. Une fois que ce même utilisateur se déplace en dors de son entreprise, il continue à se connecter via une technologie WWAN (UMTS par exemple). Lorsqu'il rentre chez lui, il trouve une connexion Wi-Fi, etc. Cette combinaison des interfaces sans fil conduit à une hiérarchie appelée *structure hiérarchisée des réseaux sans fil* (OWN : Overlay Wireless Network structure). La figure 3.1 montre un exemple d'une telle structure. Les niveaux les plus bas permettent une couverture limitée en terme de portée mais des débits importants. Les niveaux les plus hauts couvrent un nombre important d'utilisateurs sur une large zone géographique mais limitée en bande passante.

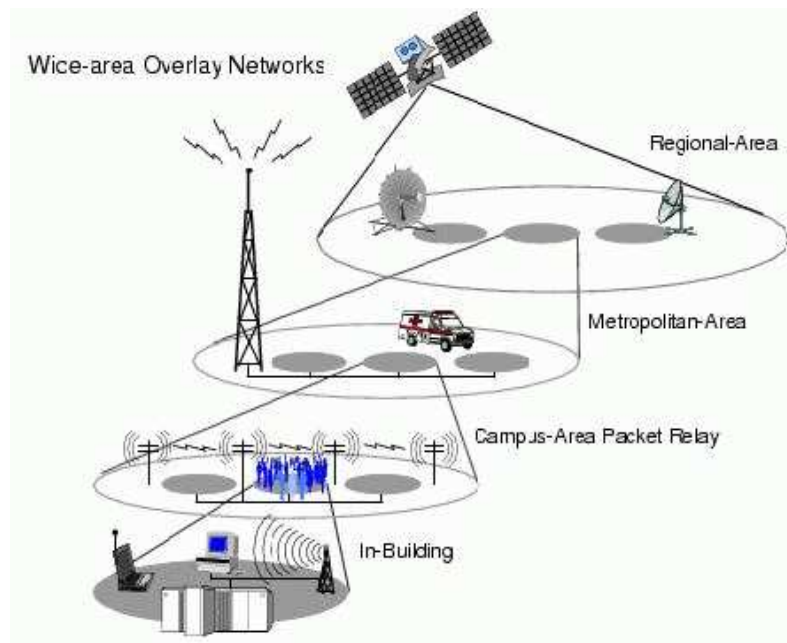


FIG. 3.1 – Un exemple d'une structure hiérarchique des réseaux sans fil.

Un bureau, un étage, une entreprise, un campus, etc, représentent en réalité des cellules de tailles différentes. Dans GSM, pour palier le manque de capacité réseau à l'explosion du nombre des abonnés, les opérateurs ont recours à différentes tailles de cellules : les macrocellules, les microcellules et les picocellules. Les macrocellules sont les plus courantes dans un réseau cellulaire GSM. Leur zone d'action s'étend jusqu'à 30 km selon les obstacles rencontrés. Les microcellules couvrent quelques rues d'un centre ville ou une station de métro (portée maximale de 500 m). Les picocellules couvrent un étage d'un grand bâtiment ou d'un centre commercial (portée maximale 100 m). Ces trois types de cellules permettent aux opérateurs de créer un réseau multicouches leur offrant ainsi plus de flexibilité pour augmenter la capacité de leur réseau. L'UMTS rajoute au sommet de cette structure multi-couches des satellites qui assurent une couverture sur l'ensemble



de la planète.

Dans une OWN, un ensemble de cellules de petite taille d'un niveau  $i$  sera regroupé dans une cellule de taille plus grande de niveau  $i + 1$ . Un exemple de regroupement de cellules dans une OWN est montré dans la figure 3.1. Il n'existe pas vraiment de classification formelle selon la taille des cellules, mais en général on peut distinguer cinq types utilisés par la communauté de communications : picocellules, nanocellules, microcellules, macrocellules, et satellites. Les cellules utilisées dans une OWN offrent plusieurs technologies différentes des réseaux mobiles et sans fil (UMTS, GSM, GPRS, IEEE 802.11, Bluetooth, Infrarouge, etc). Les caractéristiques de chaque type de cellule dans cette classification sont montrées dans la Table 3.1.

Niv.	Classification	Diamètre	débit	Latence	Mobilité	Application
0	Picocells	10m	1-50 Mbps	$\ll 10$ ms	piéton	bureau
1	Nanocells	100m	1-2 Mbps	$< 10$ ms	piéton	bureau/étage
2	Microcells	500m	100 Kbps	100 ms	piéton/véhicule	bâtiment/campus
3	Macrocells	50km	19,2 Kbps	$> 100$ ms	véhicule	métropolitain
4	Satellite	$> 50$ km	4,8 Kbps	$\gg 100$ ms	véhicule	région

TAB. 3.1: Classification des réseaux mobiles et sans fil

### 3.3 Un exemple d'intégration : 3G/WLAN

La conception et l'intégration 3G/WLAN relèvent beaucoup de défis techniques. Comme l'Authentification, l'Autorisation et l'Accounting (AAA), la sécurité globale, la Qualité de Service (QoS) et les handovers. Malgré les progrès réalisés dans la standardisation de l'interconnexion WLAN/UMTS [32,33], la plupart des efforts ont été concentrés sur l'AAA plutôt que sur la gestion de mobilité et de QoS.

L'intégration devrait être faite par étapes. L'ensemble des étapes à suivre pour d'arriver à un schéma d'interconnexion *idéal* a été proposé par 3GPP (voir [33]). Cette proposition est basée sur les services demandés par les utilisateurs. À cet égard, six scénarios ont été spécifiés, chacun représente un type différent d'interconnexion et un niveau de satisfaction de l'utilisateur. Les scénarios sont : i) scénario 1, permet d'avoir seulement des clients et une facturation commune, ii) scénario 2, prévoit le contrôle d'accès basé sur le 3GPP, iii) scénario 3, permet l'accès aux services de 3GPP PSS (*Packet-Switched Streaming*) à partir de WLANs, iv) scénario 4, assure la continuité de service après l'exécution d'un handover inter-système, v) scénario 5, promet la fonctionnalité sans couture dans le scénario précédent, et enfin, vi) scénario 6, permet d'accéder aux services du 3GPP CS (*Circuit Switching*) à partir de WLANs.

En se basant sur les scénarios ci-dessus, un certain nombre de propositions ont visé un certain degré d'intégration sans couture. Certaines d'entre elles se focalisent sur l'architecture générale et utilisent l'IP Mobile comme outil de base pour l'intégration des systèmes. Ceci permet une simple mise en œuvre avec des modifications légères aux composantes existantes de systèmes, mais en contre partie, un temps de latence de handover considérable. Ces propositions sont considérées comme des solutions *faiblement couplés* [32]. Bien qu'elles ne permettent pas d'accéder aux services 3GPP PS (*Packet Switching*), par exemple : les services WAP et MMS, à partir de WLANs, elles assurent la continuité des services (scénario 4), mais sans la fonctionnalité sans couture à tout moment (scénario 5). Les efforts de recherche dans cette catégorie de solutions se concentrent sur la conception des algorithmes d'initiation/décision pour les handovers afin d'offrir la meilleure

qualité aux utilisateurs avec une utilisation efficace des ressources.

Un autre groupe de propositions essaye d'intégrer UMTS et WLAN d'une manière plus serrée. Dans cette approche, le point d'interconnexion des deux technologies se situe soit dans le réseau cœur de l'UMTS (CN), soit dans le réseau d'accès (AN). Par conséquent, l'effort dans cette catégorie se focalise sur la résolution des problèmes techniques engendrés par les extensions ajoutées aux standards pour assurer l'interopérabilité entre l'UMTS et WLAN. La mobilité rapide et les fonctionnalités sans couture (scénario 5) sont les principales cibles de ces propositions sans oublier la complexité due aux améliorations introduites sur les composantes existantes. Ce groupe inclut les architectures *fortement couplées* [34–36] et *très fortement couplées* [37, 38].

### 3.4 Les handovers dans une OWN

Avec une OWN et d'après la figure 3.2, on distingue deux types de handovers : un handover horizontal (*horizontal handover*) et un autre vertical (*vertical handover*). Un handover horizontal est un handover classique entre deux cellules homogènes de même niveau hiérarchique dans la OWN. Par exemple, GSM vers GSM, WLAN vers WLAN, UMTS vers UMTS, UMTS vers GSM, etc. Les handovers horizontaux peuvent utiliser les algorithmes d'initiation des handovers montrés dans la section 3.1 et qui ont pour cause la mobilité des terminaux mobiles. Un handover vertical se déclenche lors du passage d'une cellule de niveau  $i$  à une autre de niveau  $i + j$  ou  $i - j$  avec  $j > 0$ , c'est-à-dire entre deux cellules hétérogènes. Par exemple, UMTS vers WLAN, GSM vers Bluetooth, etc.

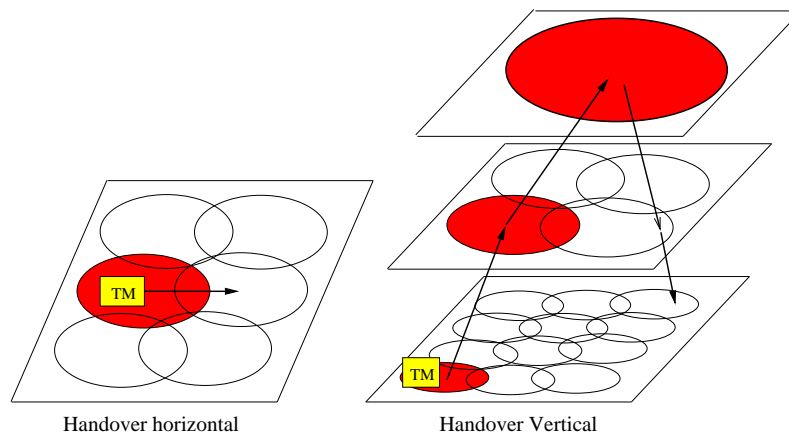


FIG. 3.2 – *Handover horizontal vs. Handover vertical.*

Il existe quelques différences importantes entre les handovers horizontaux et verticaux qui affectent les stratégies de mise en œuvre :

- Un terminal mobile procède à un handover horizontal d'une cellule  $A$  à une cellule  $B$  lorsqu'il se déplace hors de la zone de couverture de la cellule  $A$  vers celle de la cellule  $B$ . Ceci n'est pas forcément le cas pour les handovers verticaux. Le terminal mobile peut se déplacer hors de la zone de couverture d'une cellule  $A$  de niveau hiérarchique  $i$  à une zone de couverture d'une cellule  $B$  de plus grande taille de niveau  $i + 1$ , comme il peut effectuer un handover vertical de la cellule  $B$  vers la cellule  $A$  tout en restant toujours sous la couverture de la cellule  $B$  (la cellule  $B$  enveloppe la cellule  $A$ ).

- Pour pouvoir exécuter un handover vertical, le terminal mobile doit être doté de deux ou plusieurs interfaces de technologie sans fil. Chaque technologie sans fil possède ses propres caractéristiques : bande de fréquences, méthode d'accès et codages, infrastructure, etc. Par exemple, des fréquences infrarouges sont utilisées dans un bureau, alors qu'un ensemble de fréquences radios dans un bâtiment et un autre ensemble de fréquences dans un campus. Les équipements peuvent aussi utiliser différents types de couches physiques : FHSS (Frequency Hopping Spread Spectrum), DSSS (Direct Sequence Spread Spectrum), IR (Infrarouge), FDD (Frequency Division Duplex), TDD (Time Division Duplex), etc.
- Dans un réseau où les stations de base sont homogènes, le choix de la meilleure station de base est évident : le terminal mobile procède pour cela aux mesures d'un signal pilote, que chaque station de base transmet en permanence. Le signal reçu avec la meilleure qualité (selon les méthodes d'initiation des handovers) indique le meilleur point d'accès d'attachement. Dans une OWN, la puissance du signal n'est pas toujours le facteur déterminant de la meilleure station de base du réseau. Ceci est dû à la diversité des caractéristiques de chaque niveau de la OWN. Une faible puissance d'un signal provenant d'une station de base utilisant un réseau WLAN peut accomplir une meilleure performance qu'un signal puissant envoyé par une station de base qui utilise une technologie d'un réseau sans fil plus large (WWAN).

Le handover vertical est divisé en deux catégories : handover vertical montant (*upward*) et descendant (*downward*). Un handover vertical montant est un handover vertical d'une cellule de niveau  $i$  à une cellule de niveau  $i + j$ ,  $j > 0$ . Le passage à une cellule de niveau supérieur peut impliquer une dégradation de service due à la limitation de la bande passante. Un handover vertical descendant permet au terminal mobile d'accéder à une cellule de niveau  $i - j$ ,  $j > 0$  avec une bande passante supérieure à celle offerte par la cellule de niveau  $i$ . Il est logique qu'un terminal mobile *cherche sans cesse à exécuter des handovers descendants* afin d'améliorer la qualité des services de ses applications. Les handovers verticaux descendants présentent un temps moins critique que les handovers montants vu que le terminal mobile reste dans la zone de couverture de l'ancienne cellule en effectuant un handover vertical descendant alors qu'il quitte définitivement l'ancienne cellule avant de précéder à un handover vertical montant.

### 3.5 Position des objectifs

La liberté de la mobilité entre les différents réseaux sans fil dans une OWN permet aux utilisateurs de se connecter au meilleur réseau à n'importe quel instant. Cette liberté crée de nouveaux problèmes, ouvre d'autres aspects de recherche et pose de nouveaux défis. Parmi les défis on trouve : les techniques de AAA, la sécurité globale, la QoS et réservation des ressources entre les différents systèmes, la mobilité horizontale (intra-technologie) et verticale (inter-technologie).

L'objectif principal de cette thèse est de minimiser la latence des handovers verticaux tout en maintenant une consommation de la bande passante la plus basse possible. Une faible latence offre la possibilité d'un transfert souple entre les réseaux avec des coupures de communications supportables et des pertes de données négligeables. Ceci permet une meilleure interactivité entre l'utilisateur et les services multimédias offerts par les réseaux. Comme un utilisateur peut se déplacer vers une zone de mauvaise connectivité, le seul changement possible et visible pour cet utilisateur doit être lié à la limitation de la bande passante de l'interface sans fil. Par exemple, Les niveaux les plus hauts d'une OWN supportent une interactivité vidéo et audio de très bonne

qualité, alors que les niveaux les plus élevés supportent uniquement des services audio.

La puissance de consommation de la batterie est un facteur important à optimiser surtout en présence d'interfaces multiples qui peuvent fonctionner simultanément. L'approche la plus simple pour gérer les interfaces sans fil est de les laisser fonctionner en même temps. Selon des mesures présentées dans [39] faites sur des interfaces disponibles pour le grand public, les interfaces IBM Infrared WaveLAN RF [40] consomment en tout 1,5 watts lorsqu'elles sont au repos (pas de transmission ni de réception). Ce qui présente approximativement 20% de la puissance consommée par un ordinateur portable. À ce niveau de consommation, la gestion de la puissance de consommation de la batterie devient cruciale. Une solution simple consiste à faire passer périodiquement l'interface de niveau hiérarchique directement inférieur du niveau d'attachement dans la OWN en mode veille (*power saving low duty cycle sleep state*) pendant un temps  $D$  (en secondes). Les interfaces supérieures restent toujours en mode veille. Ceci est dû aux mécanismes de handover vertical où le terminal mobile cherche sans cesse à écouter les interfaces des niveaux hiérarchiques inférieurs dans l'OWN. Cette solution est très efficace et des tests expérimentaux sont montrés dans [39]. Nous utilisons cette méthode pour calculer la latence dans notre système.

L'implémentation d'un handover vertical dans une OWN utilise une partie de la bande passante qui peut être importante si la signalisation, qui est en forme des messages de handovers envoyés par le mobile ou des signaux pilotes (trames balises) envoyés par les stations de base, est importante. Les performances d'une OWN augmentent avec la réduction de la signalisation supplémentaire.

Idéalement, un utilisateur essaie de se connecter au niveau le plus bas d'une OWN, où la bande passante est très large, le plus long temps possible avant qu'il sera nécessaire de se déplacer vers un niveau plus haut. Notre système avec des cellules hétérogènes doit avoir le même comportement qu'un système de cellules homogènes en terme de la faible latence des handovers horizontaux. La raison pour exécuter un handover vertical de niveau  $i$  vers un niveau  $i + 1$  réside dans le fait que le niveau  $i$  devient indisponible en se déplaçant hors de sa zone de couverture. La découverte de l'indisponibilité de réseau d'attachement prend un certains temps afin d'assurer que le terminal mobile s'est vraiment déplacé hors de la zone de couverture. Généralement en fonction de la non réception d'un certain nombre (seuil) des trames balises envoyées par la station de base d'attachement. Ce temps augmente considérablement la latence du handover vertical et un mécanisme de prédiction serait très utile. Les mécanismes de prédiction développés dans le cadre des systèmes homogènes ne sont pas applicables dans une OWN pour les handovers verticaux. La prédiction des handovers verticaux est l'un de nos objectifs qui a un effet direct sur la minimisation de la latence.

Il est pratiquement impossible d'atteindre ces objectifs à la fois. Cependant, il faut aussi éviter les situations où la réalisation d'un objectif dégrade la réalisation des autres objectifs. Par exemple, la réduction de la puissance de consommation par la désactivation des interfaces sans fil lorsqu'ils ne sont pas utilisées augmente le temps latence. Aussi, l'émission et la réception des données sur toutes les interfaces des réseaux sans fil simultanément permettent l'obtention d'un handover avec une latence nulle mais par contre il faut s'attendre à des

consommations fortes de puissance et un gaspillage de la bande passante. Le cas idéal consiste à faire un compromis entre ces objectifs.

### 3.6 OWN : La gestion de la mobilité

Le protocole IP Mobile [8] standardisé par l'IETF [4] présente une solution incontournable pour la gestion de la mobilité sur Internet. Cette solution est bien adaptée à la macro-mobilité,

mais elle est beaucoup moins adaptée à la micro-mobilité (c'est-à-dire, mobilité inter-domaine). Nous allons proposer deux architectures (Archi-1 et Archi-2) pour deux solutions différentes mais les deux sont basées sur le protocole IP Mobile pour gérer la macro-mobilité. Dans les deux architectures, les handovers sont contrôlés par les terminaux mobiles.

### 3.6.1 Description d'Archi-1

La figure 3.3 illustre notre première architecture (Archi-1) proposée pour la mobilité IP dans une OWN. Cette architecture est basée sur différents types de technologie de réseau sans fil regroupés dans un réseau d'accès de la OWN. Ce réseau d'accès constitue un sous-domaine IP et il est attaché à Internet via une passerelle. Le déplacement d'un terminal mobile à l'intérieur d'un réseau d'accès (mobilité intra-domaine) est géré par le système sans fil correspondant (WLAN, UMTS, etc) avec un handover vertical entre les systèmes hétérogènes. Le changement de réseau d'accès (mobilité inter-domaine) est géré par le protocole IP Mobile.

L'architecture est hiérarchique dans le sens où il existe un niveau intermédiaire entre la passerelle et les différentes stations de base. Ce niveau intermédiaire est constitué d'une partie fixe sous forme d'arborescence filaire reliant la passerelle et les stations de base ou des points d'accès sans fil. La OWN est une structure virtuelle qui permet de définir le handover vertical.

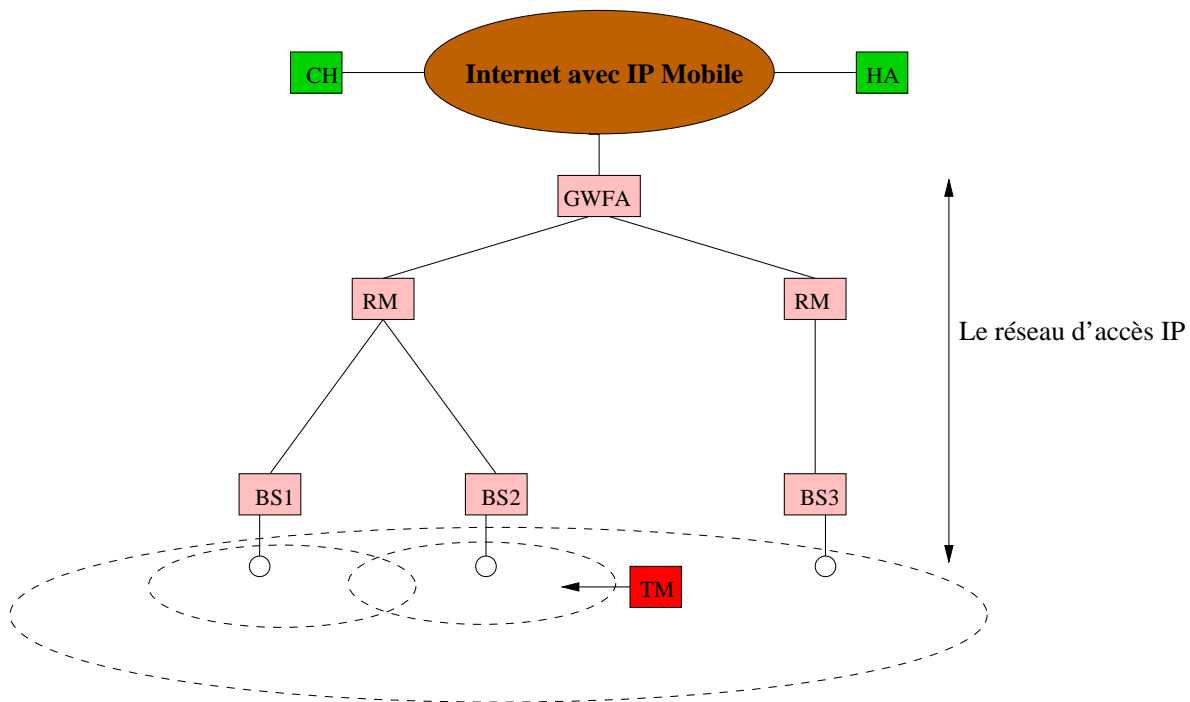


FIG. 3.3 – Archi-1 : la 1ère architecture pour la gestion de la mobilité.

L'architecture est composée de six entités fonctionnelles :

- Home Agent (HA) : routeur situé dans le réseau d'abonnement du terminal mobile. Il enregistre la localisation du terminal mobile et renvoie les paquets destinés à ce dernier vers son réseau visité.

- Correspondent Host (CH) : un nœud dans le réseau Internet qui communique avec le terminal mobile dans son réseau Home.
- Passerelle Foreign Agent (GWFA) : routeur tenant le rôle de Foreign Agent (FA) pour les nœuds visiteurs. Il tient également le rôle de HA dans le cas où le réseau d'accès est le réseau d'abonnement de certains nœuds mobiles. En outre, il implémente la fonctionnalité multicast pour router les paquets vers le terminal mobile à l'intérieur du réseau d'accès. Il diffuse périodiquement des messages d'annonce indiquant son adresse IP et il associe à chaque terminal mobile une adresse multicast unique dans son réseau d'accès.
- Routeur Multicast (RM) : routeur intermédiaire entre la GWFA et les stations de base dans le réseau d'accès. Il implémente la fonctionnalité multicast.
- Station de base (BS) : nœud reliant les parties filaires et sans fil de l'architecture. Il possède deux interfaces, une sans fil pour communiquer avec les terminaux mobiles et l'autre filaire pour communiquer avec la passerelle via l'arborescence filaire.
- Terminal Mobile (TM) : un nœud dans le réseau d'accès possédant plusieurs interfaces sans fil. Il implémente la procédure d'enregistrement IP Mobile. Cette procédure est déclenchée lors d'un changement de réseau d'accès.

### 3.6.2 Fonctionnement d'Archi-1

Lorsqu'un TM est dans son réseau d'abonnement, un CH peut communiquer avec lui sur Internet via le routage IP normal. Une fois ce TM se déplace vers un réseau visiteur, le protocole IP Mobile intervient pour permettre au TM d'obtenir une COA par la GWFA du réseau d'accès auquel il est attaché et de s'enregistrer auprès de son HA. En plus, le TM doit acquérir une adresse IP multicast unique dans le réseau d'accès proposée par la GWFA. Un petit groupe de BSs est sélectionné par le TM pour écouter cette adresse multicast. Ce groupe forme le groupe multicast associé au TM. La raison de la sélection d'un groupe multicast de stations de base réside dans le fait que ces stations de base sont voisines et lorsque le TM se déplace, il va s'y trouver couvert par une des stations de son groupe. Une BS dans ce groupe est sélectionnée par le TM comme une *forwarding* BS, elle relaie immédiatement les paquets multicast envoyés par la GWFA. Les autres BSs du groupe multicast jouent le rôle de *buffering* BSs, elle stockent les paquets multicast envoyés par la GWFA dans un *buffer* circulaire afin de maintenir les paquets les plus récents et d'éviter le surchargement des *buffers*. La bufferisation des paquets par les BSs voisines de la *forwarding* BS permet au TM de récupérer les paquets non reçus lors de la procédure de handover.

La GWFA décapsule les paquets envoyés par le HA vers le TM. Elle utilise l'adresse multicast associée au TM correspondant pour envoyer les paquets en mode multicast.

Les BSs envoient périodiquement des trames balises pour annoncer leurs présences et relayer d'autres informations. Le TM écoute en permanence ces trames et détermine la *forwarding* BS, les *buffering* BSs et les BSs qui ne font pas partie de son groupe multicast. Lorsque le TM initialise un handover, il demande à l'ancienne BS de passer de l'état *forwarding* BS à l'état *buffering* BS. Les paquets stockés par la nouvelle *forwarding* BS et qui ne sont pas encore reçus par le TM sont renvoyés à ce dernier.

### 3.6.3 Les handovers dans Archi-1

Dans les cellules homogènes (les cellules de même niveau hiérarchique), les puissances des signaux des trames balises reçues par le TM sont comparées et la BS avec le signal le plus puissant (selon les mécanismes d'initiation des handovers montrés dans la section 3.4) est sélectionnée comme la *forwarding* BS. La figure 3.4 montre en détail le mécanisme du handover horizontal. En plus des envois périodiques de trames balises, la *forwarding* BS relaie les paquets de données vers le TM. Si le TM reçoit une trame balise d'une BS de meilleure qualité que celle de *forwarding* BS, il décide d'initier un handover horizontal en demandant à la nouvelle BS d'arrêter la bufferisation et de se comporter comme une *forwarding* BS. À l'ancienne BS, il demande de passer de l'état de *forwarding* BS à l'état *buffering* BS. La latence d'un handover horizontal commence dès l'instant où le TM décide que le signal reçu de la nouvelle BS est meilleur que le signal de l'ancienne BS jusqu'à la réception du premier paquet de données provenant de la nouvelle BS.

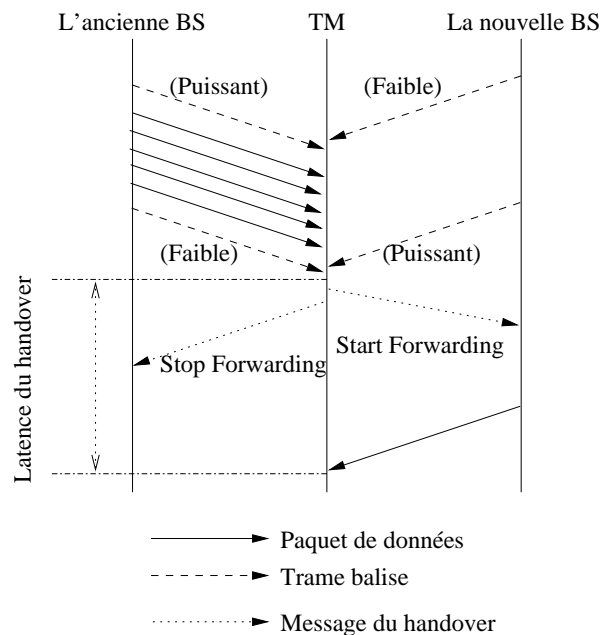


FIG. 3.4 – Le mécanisme du handover horizontal dans l'Archi-1.

Entre les cellules hétérogènes (les cellules de différents niveaux hiérarchiques), le TM écoute les différentes trames balises et choisit le meilleur signal non pas seulement en fonction de sa puissance mais aussi en fonction d'autres métriques (la qualité du signal, le débit offert, etc). La BS avec le meilleur signal est sélectionnée comme une *forwarding* BS et un petit ensemble des BSs est considéré comme *buffering* BSs. Un handover vertical montant est initialisé lorsqu'un certain nombre de trames balises (selon un seuil  $T_B$ ) n'a pas été reçu du niveau hiérarchique d'attachement de la OVN (le niveau auquel appartient la *forwarding* BS). Le TM décide que le niveau hiérarchique d'attachement est inaccessible et il effectue un handover vertical montant vers un niveau accessible directement supérieur dans la OVN. Dans ce cas, le TM demande à la *buffering* BS de meilleur signal d'un niveau hiérarchique supérieur de passer à l'état *forwarding* BS. Comme l'ancienne *forwarding* BS n'est plus accessible directement, le TM envoie une requête à cette BS pour qu'elle cesse la transmission via la nouvelle *forwarding* BS. Ce procédé est montré dans la figure 3.5. Les croix sur cette figure signifient les points logiques de l'atténuation des

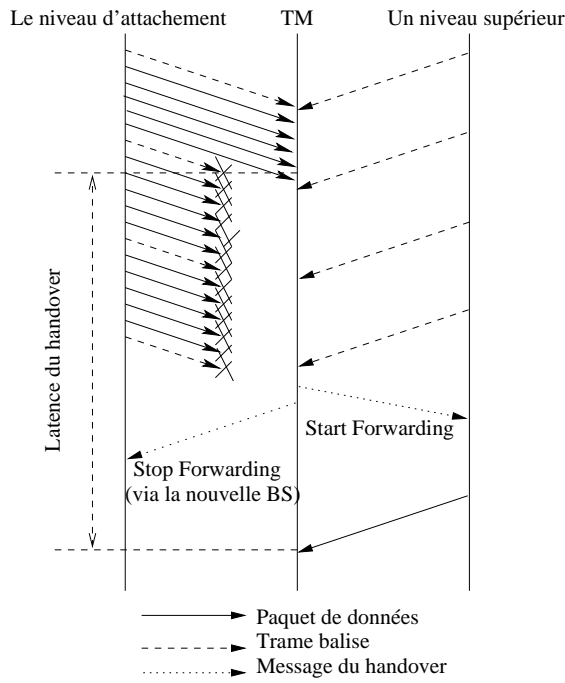


FIG. 3.5 – Le mécanisme du handover vertical montant dans l'Archi-1.

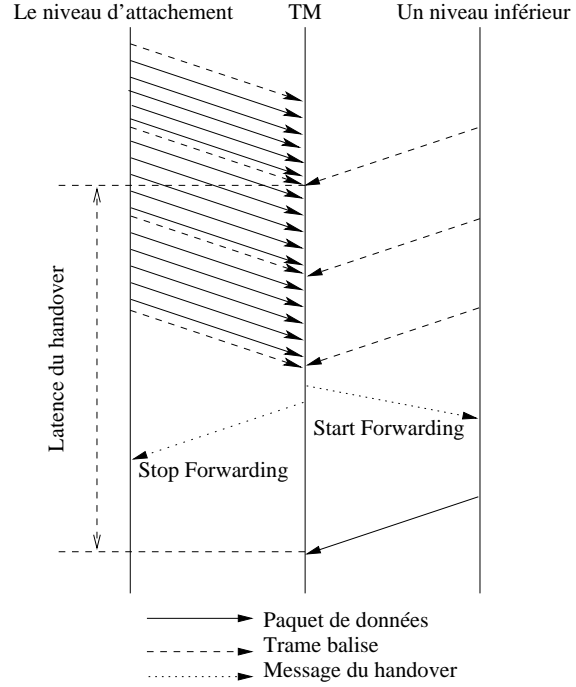


FIG. 3.6 – Le mécanisme du handover vertical descendant dans Archi-1.

messages et non pas les chemins empruntés par ces messages. La latence d'un handover vertical montant est donnée par la durée entre le moment où le niveau hiérarchique

d'attachement de la OWN commence à devenir inaccessible jusqu'à la réception du premier paquet de données envoyé par la nouvelle BS.

Lorsqu'un certain nombre de trames balises (selon un seuil  $T_B$ ) est reçu successivement d'un niveau hiérarchique inférieur dans la OWN, le TM effectue un handover vertical descendant (figure 3.6). La latence d'un handover vertical descendant est donnée par la durée entre le moment où un niveau hiérarchique inférieur dans la OWN commence à devenir accessible jusqu'à la réception du premier paquet de données envoyé par la nouvelle BS.

### 3.6.4 Latence et overhead dans Archi-1

La latence d'un handover vertical  $L$  est définie par le temps nécessaire pour qu'un TM puisse se déconnecter de l'ancienne BS et recevoir le premier paquet de la nouvelle BS.  $L$  est donnée pour la formule suivante :

$$L = L_D + L_P + L_N + L_F$$

- $L_D$  est le temps nécessaire pour qu'un TM découvre qu'il doit se connecter à un nouveau niveau hiérarchique de la OWN. En absence de  $T_B$  trames balises du niveau d'attachement, le TM décide que ce niveau est inaccessible et il doit initier un handover vertical montant vers un niveau plus haut accessible dans la OWN. La réception de  $T_B$  trames balises d'un niveau inférieur dans la OWN donne au TM la possibilité d'effectuer un handover vertical descendant.  $L_D$  dépend fortement de la fréquence d'émissions des trames balises par les



BSs. Si l'espace de temps entre deux trames balises est large, la valeur de  $L_D$  sera grande par le fait que le TM prendra un temps important pour découvrir l'accessibilité d'un niveau inférieur ou l'inaccessibilité du niveau courant d'attachement de la OWN. Soit  $N_B$  l'espace de temps entre deux trames balises (en secondes). La borne supérieure de  $L_D$  est donnée par  $N_B \times T_B$ . La borne inférieure est égale à  $N_B \times (T_B - 1)$ . Par conséquent, la moyenne est donnée par  $N_B \times (T_B - 1) + N_B/2$ . Cette composante de la latence est invisible comme une déconnexion pour les utilisateurs dans le cas d'un handover horizontal ou vertical descendant puisque le TM reste toujours connecté à l'ancienne BS pendant qu'il découvre qu'il est possible de se connecter à une nouvelle BS. Si on considère  $D$  (en secondes) le temps périodique de mode en veille de l'interface de niveau hiérarchique directement inférieur au niveau d'attachement, alors il faut rajouter à la latence moyenne  $L_D$  des handovers verticaux descendants un temps moyen de  $D/2$ .

- $L_P$  est la latence nécessaire pour qu'un TM effectue une transition vers l'interface qui correspond à la technologie utilisée par la nouvelle BS. Cette composante peut prendre la valeur 0 si l'état de l'interface cible est déjà en marche lors de l'apparition du handover.
- Le TM demande à la nouvelle BS de se comporter comme une *forwarding* BS en un temps  $L_N$ . Ceci dépend généralement de la latence entre le TM et la BS.  $L_N$  prend un temps  $L_U + S_M/Bw_U$  secondes pour les handovers verticaux montants et  $L_L + S_M/Bw_L$  pour les handovers verticaux descendants. ( $L_U, Bw_U$ ) et ( $L_L, Bw_L$ ) représentent respectivement (la latence (en secondes), La bande passante (en bits/s)) de la technologie d'interface cible du niveau supérieur et inférieur dans la OWN.  $S_M$  (en bits) représente La taille de message de changement d'état envoyé par le TM à la nouvelle BS.
- $L_F$  est la temps nécessaire pour que la nouvelle BS puisse envoyer le premier paquet de données vers le TM.  $L_F = 0$  s'il n'y a aucun paquet à envoyer.  $L_F$  prend un temps  $L_U + S_{Data}/Bw_U$  secondes pour les handovers verticaux montants et  $L_L + S_{Data}/Bw_L$  pour les handovers verticaux descendants.  $S_{Data}$  est la taille de ce premier paquet destiné au TM.

$L_D$  et  $L_P$  se chevauchent dans le cas où le TM peut prédire dans le temps l'inaccessibilité du niveau courant d'attachement dans la OWN et il met en marche à l'avance l'interface d'un niveau supérieur accessible.  $L_D, L_N$  et  $L_F$  peuvent aussi se chevaucher si la nouvelle BS est déjà dans l'état *forwarding* BS vers le TM. Ce cas est possible dans l'optimisation montrée dans la section 3.7.3 où le TM ordonne plusieurs BSs de se comporter comme *forwarding* BSs. Par contre  $L_P$  et  $L_N$  ne peuvent jamais se chevaucher.

L'overhead  $B$  est le nombre de bits de la signalisation envoyée par seconde. Les trames balises et d'autres messages que le TM utilise pour initier un handover sont considérés comme un overhead. La valeur moyenne de l'overhead pour un handover vertical montant ou descendant est  $(1/N_B)S_B$ , où  $S_B$  est la taille (en bit) d'une trame balise.

### 3.6.5 Description d'Archi-2

Dans Archi-1, le TM ne s'attache pas forcément à toutes ses *buffering* BSs ce qui engendre des bufferisations inutiles. Afin d'éviter ce problème, l'architecture 2 (Archi-2) remplace les routeurs multicast (RM) dans le réseau d'accès d'Archi-1 par des routeurs IP normaux. Le but de cette architecture est d'éviter l'utilisation de la technique de groupe multicast qui bufferise les paquets de données de chaque TM. La figure 3.7 illustre notre deuxième architecture (Archi-2).

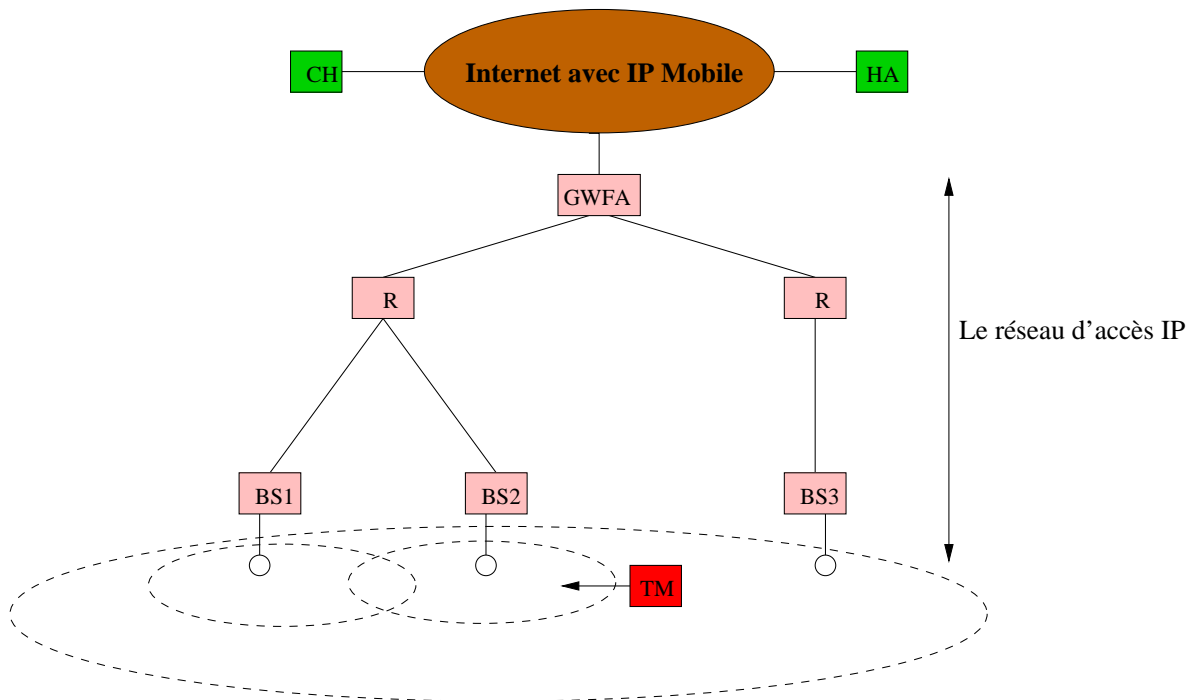


FIG. 3.7 – Archi-2 : la 2ème architecture pour la gestion de la mobilité.

### 3.6.6 Fonctionnement d'Archi-2

Dans cette architecture et contrairement à Archi-1, le TM visiteur doit acquérir seulement une COA par la GWFA du réseau d'accès auquel il est attaché et s'enregistrer auprès de son HA. Le TM recherche constamment la BS qui offre le meilleur signal. Une fois attaché à une cellule, la BS de cette cellule relaie les paquets de données vers lui. Il n'y a pas de bufferisation par les cellules voisines dans cette architecture. Pour assurer une perte de paquets nulle lors des handovers, la BS où le TM est attaché stocke les derniers paquets de données envoyés. Ces paquets font l'objet d'une retransmission vers la nouvelle BS sélectionnée par le TM. Cette retransmission est effectuée par l'ancienne BS via le réseau filaire vers la nouvelle BS. Le fonctionnement d'IP Mobile est préservé dans Archi-2.

### 3.6.7 Les handovers dans Archi-2

À la réception d'une trame balise de meilleure qualité d'une cellule voisine homogène (même technologie sans fil), le TM initie un handover horizontal en suivant les étapes suivantes (figure 3.8) :

- Le TM envoie un message *Greet* (un message pour initialiser le handover) à la nouvelle BS. Ce message contient l'adresse de TM, l'adresse de l'ancienne BS et une petite liste des *IDentificateurs* (IDs) des derniers paquets de données reçues par le TM.
- La nouvelle BS crée une nouvelle entrée dans sa table de routage pour le TM. Cette entrée représente la capacité de la nouvelle BS de relayer les données vers ce TM. La nouvelle BS acquitte le message *Greet* en envoyant un *Greet Ack* au TM. À la réception de l'acquiescement, le TM est attaché définitivement à la nouvelle BS. Pour éviter la perte des données

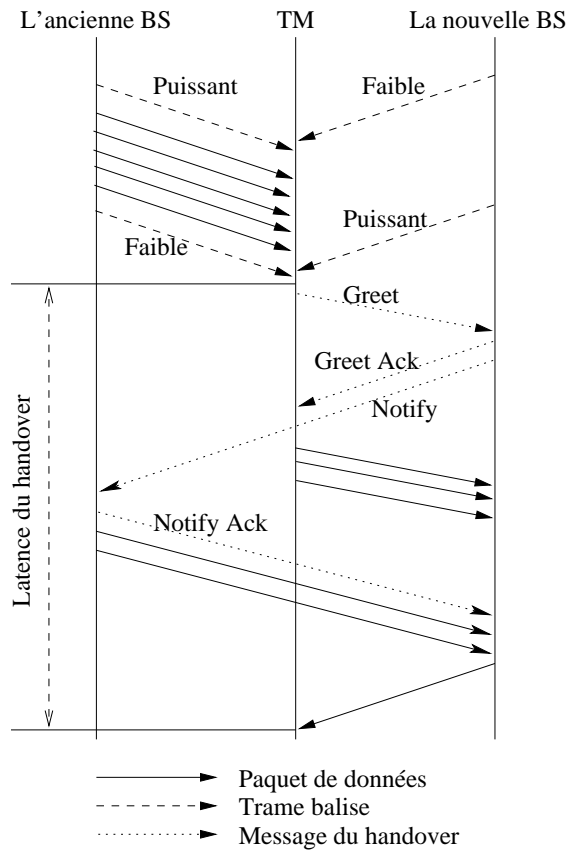


FIG. 3.8 – Le mécanisme du handover horizontal dans Archi-2.

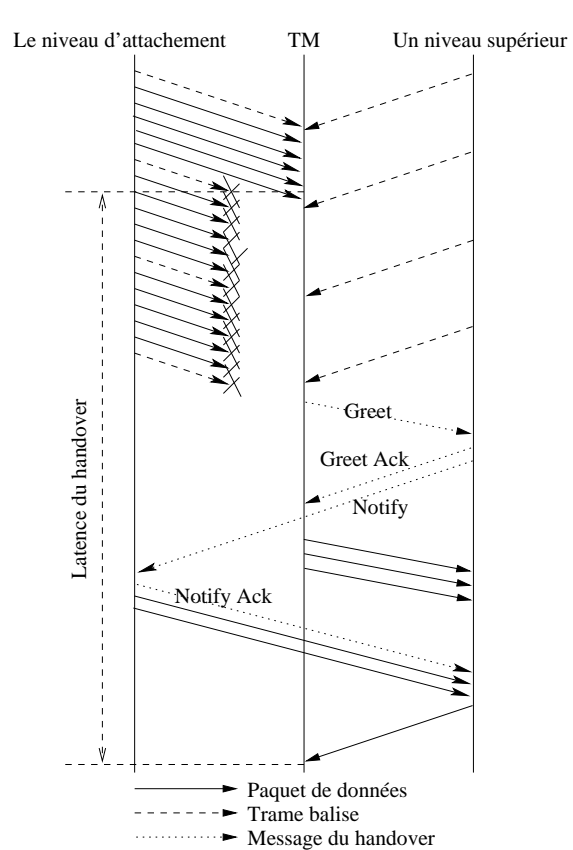


FIG. 3.9 – Le mécanisme du handover vertical montant dans Archi-2.

envoyées récemment à l'ancienne BS, et qui sont dans le buffer de TM, ces données sont renvoyées à la nouvelle BS.

- L'ancienne BS est informée que le TM a changé la cellule par un message de notification (*Notify*) envoyé via le réseau filaire par la nouvelle BS. Ce message contient aussi la liste des identificateurs des paquets reçus récemment par le TM. Cette liste est la même liste transmise précédemment par le TM dans le message *Greet* à la nouvelle BS. Cette technique permet à l'ancienne BS de retransmettre les paquets de données qui ne sont pas reçus par le TM lorsqu'il était attaché à sa cellule en passant par la nouvelle BS.
- À la réception de la notification, l'ancienne BS procède à une mise à jours de sa table de routage en retirant l'entrée correspondante à ce TM. Elle envoie les paquets de données disponibles dans son buffer dont les identificateurs ne figurent pas dans le message de notification vers la nouvelle BS. Elle répond aussi par un *Notify Ack*.

Ce processus du handover épargne les BSs des cellules voisines d'effectuer des bufferisations qui peuvent être inutiles. L'échange des messages de notification entre la nouvelle BS et l'ancienne BS permet d'assurer la retransmission des données qui peuvent être perdues durant le handover.

Le principe des handovers verticaux montant est illustré dans la figure 3.9. Les messages *Greet*, *Greet Ack*, *Notify* et *Notify Ack* sont décrit précédemment.

### 3.6.8 Latence et Overhead dans Archi-2

le temps latenc  $L$  des handovers dans cette Archi-2 est

$$L = L_D + L_P + L_{Greet} + L_{Notify} + L_T + L_N$$

- $L_{Greet}$  est le délai entre la transmission et la réception d'un message *Greet* envoyé par le TM à sa nouvelle BS.  $L_{Greet}$  prend un temps  $L_U + S_{Greet}/Bw_U$  secondes pour les handovers verticaux montants et  $L_L + S_{Greet}/Bw_L$  pour les handovers verticaux descendants.  $S_{Greet}$  est la taille (en bits) de message *Greet*.
- $L_{Notify}$  est la latence d'envoyer un message *Notify* par la nouvelle BS à l'ancienne BS. Cette latence dépend de la taille de message *Notify* ( $S_{Notify}$ ), du nombre de liens ou sauts ( $H$ ) séparant l'ancienne et la nouvelle BS dans le réseau filaire, de la latence ( $L_W$ ) et la bande passante ( $Bw_W$ ) de chaque routeur dans le réseau filaire qui connecte les deux BSs. La valeur de  $L_{Notify}$  est donnée par  $[(S_{Notify}/Bw_W) + L_W] \times H$ .
- $L_T$  représente le temps entre la transmission du premier paquet de données dont l'identificateur ne fait pas partie de la liste des *IDs* par l'ancienne BS à la nouvelle BS passant par le réseau filaire.  $L_T$  prend  $[(S_{Notify}/Bw_W) + L_W] \times H$ .

La latence  $L$  de cet algorithme de handover est quasiment équivalente à celle présentée dans l'Archi-1 en considérant que la retransmission des données entre l'ancienne BS et la nouvelle BS sur le réseau filaire est négligeable vu que les deux BSs ne sont pas très éloignés l'une de l'autre et le réseau filaire présente un débit et fiabilité plus importants. L'overhead de la bande passante dépend seulement de la fréquence d'émission des trames balises. La valeur moyenne de l'overhead pour un handover vertical montant ou descendant est  $(1/N_B)S_B$ , où  $S_B$  est la taille (en bit) d'une trame balise.

## 3.7 Optimisation des handovers

Les applications temps réels tels que la voix ou la vidéo nécessitent un temps de latence très faible lors des handovers afin d'assurer la transparence auprès des utilisateurs. Dans la littérature, plusieurs méthodes d'optimisation des handovers classiques ont été développées. Ces méthodes peuvent être toujours utilisées lorsqu'un terminal mobile change sa cellule d'attachement en restant dans le même niveau hiérarchique dans la OWN. Les handovers verticaux descendants offrent aux TMs l'avantage de rester connectés à l'ancienne BS durant le processus du transfert vers la nouvelle BS. Par conséquent, l'optimisation de la latence des handovers verticaux descendants ne présente pas une priorité. Les handovers verticaux montants consomment beaucoup de temps avant qu'un TM réalise qu'il est hors de sa cellule d'attachement et qu'il est nécessaire de remonter dans les niveaux hiérarchiques de la OWN. Les composantes  $L_D$  et  $L_P$  de la latence  $L$  sont les seuls paramètres qui offrent la possibilité d'une optimisation. L'intervalle de temps avant d'initier un handover vertical montant et qui consiste en l'absence de  $T_B$  trames balises de l'ancienne BS peut être contourner par deux méthodes : la prédiction ou l'augmentation de la fréquence d'émission des trames balises par les BSs. La latence  $L_P$  dépend de la manière de faire fonctionner l'ensemble des interfaces d'un TM.

### 3.7.1 La cellule virtuelle

On a vu précédemment que les BSs envoient périodiquement des trames balises. Chaque TM conserve la trace de la puissance du signal reçu appelée RSS (Received Signal Strength) de sa BS d'attachement. Lorsque la RSS de cette BS commence à s'affaiblir au-delà d'un certain seuil, le TM commence à mesurer les RSSs des cellules voisines. L'initiation d'un handover horizontal est due à la réception d'une trame balise avec une RSS supérieure à la RSS de la BS d'attachement tandis qu'un handover vertical est initié dès l'absence ou la réception de  $T_B$  trames balises. On a vu aussi que la latence d'un handover vertical montant augmente avec l'augmentation de l'espace de temps séparant la transmission des trames balises. Ceci est dû au temps d'attente des  $T_B$  trames balises. Le concept des cellules virtuelles permet d'éliminer ce temps d'attente.

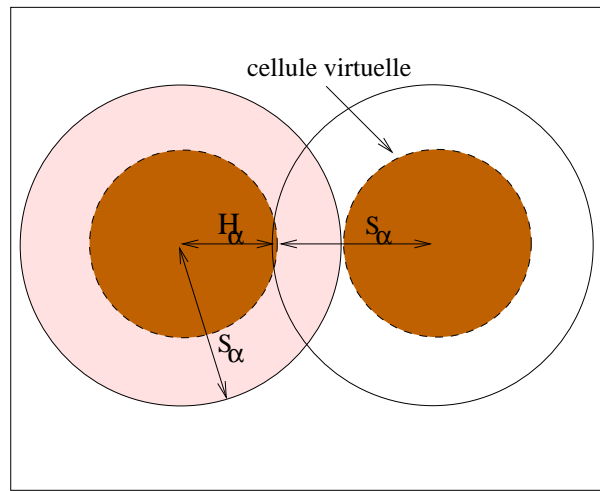


FIG. 3.10 – La technique de la cellule virtuelle.

On définit deux paramètres :  $S_\alpha$  et  $H_\alpha$ . Le seuil d'une puissance de signal acceptable ( $S_\alpha$ ) est la borne inférieure de la RSS d'une BS sous laquelle une communication devient impossible. Ce seuil forme les frontières d'une cellule. Le seuil  $H_\alpha$  est la borne inférieure de la RSS de la BS d'attachement dans laquelle la RSS d'une BS voisine est supérieure ou égale à  $S_\alpha$ . Un TM dans une cellule virtuelle reçoit des trames balises de la BS d'attachement avec une RSS supérieure à  $H_\alpha$ . Chaque cellule possède sa propre cellule virtuelle. La figure 3.10 montre les deux paramètres qui permettent d'approuver la présence des cellules virtuelles. Un TM en dehors de la cellule virtuelle de la BS d'attachement doit recevoir des trames balises des cellules homogènes. Si ce n'est pas le cas, le TM procède à un handover vertical montant lorsque la RSS de la BS d'attachement devient inférieure à  $S_\alpha$  sans compter l'absence de  $T_B$  trames balises.

La technique de la cellule virtuelle permet d'annuler la composante  $L_D$  de la latence pour les handovers verticaux montants. Les paramètres  $S_\alpha$  et  $H_\alpha$  sont faciles à fixer pour chaque type de cellule de la technologie sans fil utilisée. Ces paramètres sont calculés en fonction de la puissance de transmission de la BS d'attachement, des BSs voisines, de l'environnement et de la bande passante offerte par la technologie.

### 3.7.2 Émission rapide des trames balises

Le TM peut demander à la BS d'attachement (dans Archi-2) ou à un sous-ensemble de son groupe multicast (dans Archi-1) d'augmenter la fréquence d'émissions des trames balises et donc la réception des trames balises sera plus rapide. Cette technique permet de réduire la latence  $L_D$ . La figure 3.11 montre le gain en termes de latence  $L_D$ . Les autres composantes de temps latence  $L$  sont identiques à celles de la méthode classique.

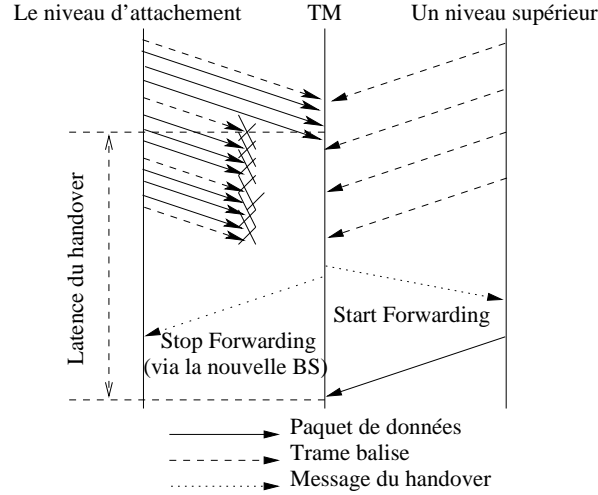


FIG. 3.11 – *Emission rapide des trames balises.*

Cette technique a l'inconvénient d'ajouter un coût supplémentaire en terme de consommation de puissance et d'overhead.

### 3.7.3 Diffusion multiple des paquets (DMP)

Cette technique concerne la procédure des handovers verticaux dans Archi-1. Le TM peut demander à plusieurs BSs de son groupe multicast de relayer ses paquets de données au lieu d'une seule *forwarding* BS. Ceci implique la réception de plusieurs copies de chaque paquet de données (figure 3.12). Afin de minimiser la réception multiple des paquets tout en optimisant la latence des handovers verticaux montants, le TM ordonne à deux BSs de se comporter comme *forwarding* BSs. La première *forwarding* BS est sélectionnée selon la procédure classique et la seconde est une BS d'un niveau hiérarchique supérieur de la première *forwarding* BS et qui possède le meilleur signal. La duplication des paquets de données reçus est filtrée au niveau couche IP des TMs en conservant dans un cache l'ID de chaque paquet récemment reçu. La couche IP des TMs conserve aussi la trace des paquets reçus par chaque interface. Lorsque  $T_D$  consécutifs paquets de données sont reçus par la nouvelle interface et non pas par l'ancienne interface, le TM décide que l'ancien niveau de la OWN est inaccessible et il initie un handover vertical montant vers la nouvelle interface.

Le TM doit recevoir  $T_D$  paquets de données (ou  $T_B$  trames balises si aucun paquet de données n'est envoyé) par une de ses interfaces supérieures avant l'initiation d'un handover vertical montant. Dans ce cas,  $L_D$  prend un temps  $N_D(T_D - 1) + N_D/2$  secondes où  $N_D$  est l'espace de temps entre la transmission de deux paquets de données. Il est clair que les autres composantes,  $L_N$  et  $L_F$ , n'existent pas vu que la nouvelle BS est déjà une *forwarding* BS. Cependant, cette approche

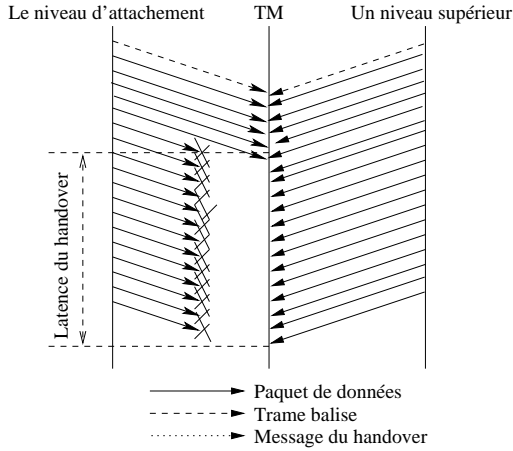


FIG. 3.12 – Diffusion multiple des paquets.

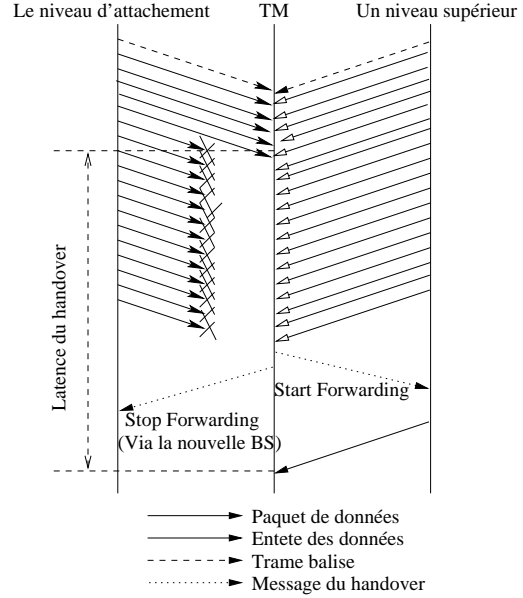


FIG. 3.13 – Diffusion multiple des entêtes.

présente une consommation supplémentaire de la puissance due au fonctionnement simultanée de deux interfaces. L'overhead compte, en plus des trames balises, les copies de paquets de données envoyées par la BS de niveau hiérarchique supérieur dans la OVN. Au moyenne on peut avoir  $(1/N_B)S_B + (1/N_D)S_D$  b/s. Cette technique ne peut fonctionner que si les deux niveaux de la OVN sont capable de supporter le même chargement de bande passante.

### 3.7.4 Diffusion multiple des entêtes (DFE)

Dans la méthode précédente, une BS d'un niveau hiérarchique supérieur dans la OVN se comporte comme une *forwarding* BS en même temps que la BS d'attachement. Avec la technique de diffusion multiple des entêtes, la même BS de niveau hiérarchique supérieure dans la OVN bufférisse les paquets de données comme dans la méthode classique mais par contre elle envoie uniquement les entêtes IP des paquets (figure 3.13). Le cache de la couche réseau de TM conserve la trace des paquets ainsi les entêtes IP reçus de chaque interface. Ce TM bascule sur la nouvelle interface lorsque  $T_D$  entêtes IP sont reçues par la nouvelle interface alors qu'aucun paquet n'est reçu par l'ancienne interface. Cette approche permet de réduire l'overhead généré par la diffusion multiple des paquets.

Etant donnée que le nouvelle BS était en état de *buffering* BS, alors contrairement à l'approche de diffusion multiple des paquets, les composantes de latence  $L_N$  et  $L_F$  ont les mêmes valeurs que la méthode de base. C'est-à-dire,  $L_N = (L_U + S_M/B_U, L_L + S_M/B_L)$  et  $L_F = (L_U + S_D/B_U, L_L + S_D/B_L)$  pour le handover vertical (montant, descendant) respectivement. La puissance de consommation de la batterie d'un TM est la somme des consommations des deux interfaces qui fonctionnent en même temps tandis que l'overhead dépend fortement de la taille et du nombre des trames balises sans oublier le nombre des entêtes IP envoyés. La valeur moyenne de l'overhead est donnée par  $((1/N_B)S_B + (1/N_D)S_H)$  b/s où  $S_H$  est la taille des entêtes IP en bits).

### 3.8 Les handovers inter-domaines

Afin d'éviter le coût d'exécution d'IP Mobile à chaque changement d'un réseau visiteur (le domaine de la GWFA), une chaîne de retransmission entre les GWFAs est utilisée. Cette chaîne est maintenue par une signalisation supplémentaire entre la nouvelle GWFA et l'ancienne. Lorsqu'un TM se déplace d'un réseau visiteur à un autre, la nouvelle GWFA informe l'ancienne GWFA de la localisation de ce TM dont le but de relayer les paquets de données destinés au TM vers la nouvelle GWFA (figure 3.14). Au fur et à mesure que le TM traverse différents domaines GWFA, une chaîne de retransmission est formée entre les GWFA. Lorsque cette chaîne atteint une certaine

limite, IP Mobile est appliqué à nouveau entre le réseau visiteur actuel et le HA. Par conséquent, la chaîne est brisée et la nouvelle GWFA devient le début d'une possible nouvelle chaîne.

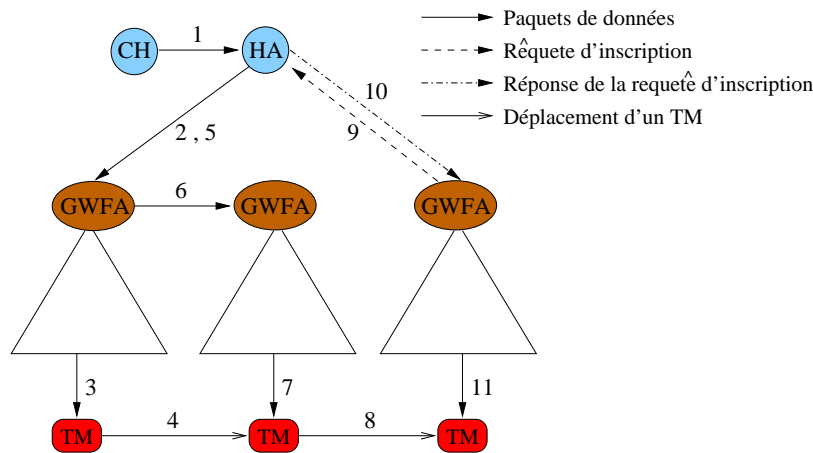


FIG. 3.14 – Un exemple d'une chaîne de retransmission.

### 3.9 La mobilité rapide et faible

Les cellules de chaque niveau hiérarchique de la OVN supportent une mobilité raisonnable imposée par l'environnement (bureau, étage, campus, région, etc). Les Picocells et Nanocells sont en charge de la couverture intérieure (*indoor*) à faible mobilité, tandis que les Macrocells et Satellite assurent une couverture des environnements extérieure (*outdoor*) desservant des mobiles qui se déplacent à grande vitesse. Les Microcells peuvent supporter une mobilité rapide dans un environnement extérieur ou une mobilité faible dans un environnement intérieur.

Nous proposons deux classes de mobilité. La mobilité rapide, représentée généralement par les TMs qui se déplacent dans un véhicule ; et la mobilité faible des piétons. Les TMs avec une mobilité rapide nécessitent plus de handovers et signalisations dans les niveaux hiérarchiques les plus bas de la OVN. Dans ce cas et afin de minimiser le nombre de handovers, les niveaux hiérarchiques supérieurs de la OVN où les cellules couvrent une zone géographique plus large sont adaptables à la mobilité rapide.

La figure 3.15 montre le procédé de la transition entre les deux classes. Une vitesse seuil (*speed-threshold*) est associée à chaque niveau hiérarchique de la OVN qui limite la vitesse maximale des utilisateurs ou le nombre des handovers dans un intervalle de temps. Il est clair qu'un niveau  $i + 1$  nécessite une vitesse seuil supérieur à celui d'un niveau  $i$ . Initialement, chaque TM



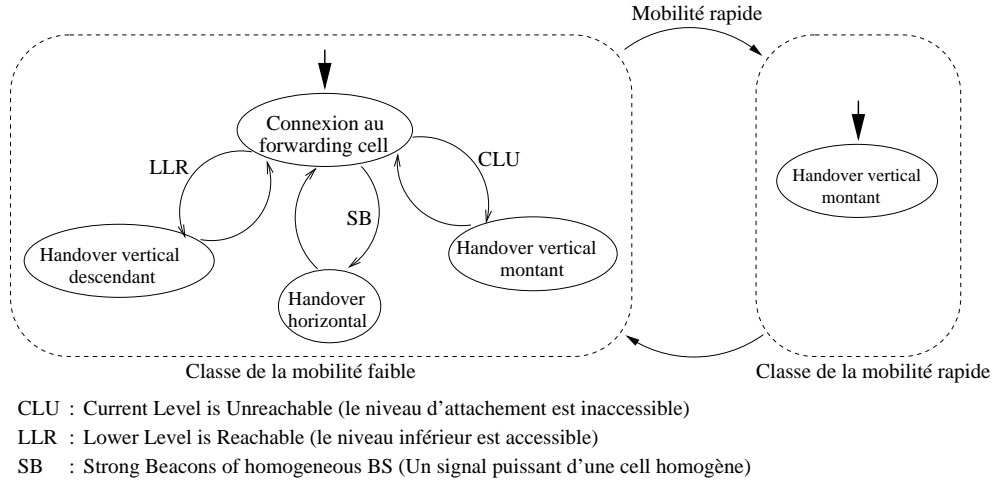


FIG. 3.15 – La gestion de la mobilité rapide.

se considère dans la classe où la mobilité est faible. Dans cet état, le TM fonctionne normalement en exécutant des handovers selon les algorithmes présentés dans la section 3.6. Lorsque ce TM détecte qu'il se déplace à une vitesse plus grande que le seuil autorisé par le niveau hiérarchique d'attachement, il change son état d'une mobilité faible à une mobilité rapide et procède à un handover vertical montant. Ensuite il retourne à l'état de la faible mobilité.

### 3.10 Évaluation des performances

Dans cette section, les performances de notre système de handovers sont étudiées par des simulations sous OPNET [41]. Le simulateur OPNET offre plusieurs outils pour le développement des modèles, l'exécution de la simulation et l'analyse des résultats.

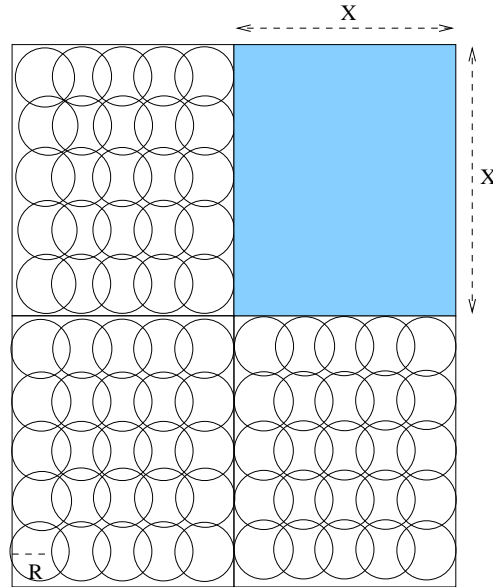
#### 3.10.1 Le modèle de simulation

Une GWFA est placée au sommet de notre architecture de simulation pour envoyer des paquets de donnée en multicast ou unicast vers les TMs selon le type d'architecture utilisé (Archi-1 ou Archi-2). Nous avons considéré Deux niveaux hiérarchiques qui permettent aux TMs d'avoir deux interfaces de technologies sans fil.

#### Le modèle topologique

La OWN contient deux niveaux hiérarchiques (figure 3.16). Le niveau le plus bas (niveau 0) consiste en une collection de cellules identiques de format circulaire et de rayon  $R$  qui offrent une bande passante très élevée. Le niveau supérieur (niveau 1) consiste en un ensemble de cellules de format carrée de côté  $X$  qui couvrent une large zone géographique avec une faible bande passante.  $T\%$  des cellules carrées contiennent des cellules circulaires identiques et uniformément distribuées. Le nombre des cellules circulaires est suffisant pour couvrir toute la surface d'une cellule carrée. Si  $T = 50$ , la moitié des cellules carrées comportent des petites cellules circulaires.

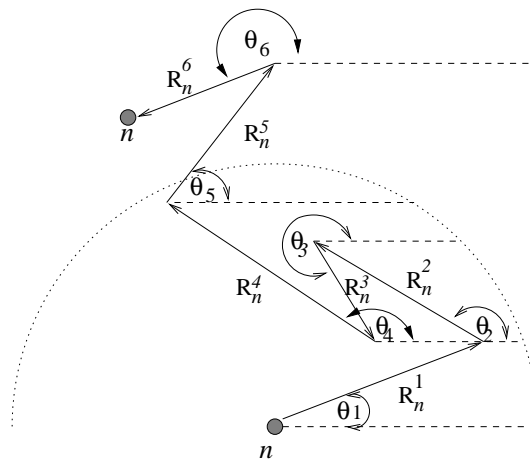
Les stations de base sont modélisées par des files d'attente et placées au centre des cellules correspondantes (cercle ou carré). Le nombre maximal des TMs (*Nombre\_max\_mobiles*) est l'un

FIG. 3.16 – *Le modèle topologique.*

des paramètres de simulation. Chaque TM est placé aléatoirement dans la zone de déplacement par la sélection aléatoire de ses coordonnées  $x$  et  $y$ .

### Le modèle de mobilité

Le modèle de mobilité [42] proposé est un processus stochastique continu dans le temps, qui caractérise les mouvements des nœuds dans un espace à deux dimensions. Les mouvements de chaque nœud consistent à une séquence d'intervalles de taille aléatoire, durant lesquels la direction et la vitesse sont constantes. La vitesse et la direction varient aléatoirement d'un intervalle à un autre. Par conséquent, durant un intervalle  $i$  de durée  $T_n^i$ , le nœud  $n$  parcourt une distance de  $V_n^i T_n^i$  sur la même ligne avec un angle  $\theta_n^i$ . La figure 3.17 illustre le mouvement d'un nœud  $n$  sur six intervalles de mobilité.

FIG. 3.17 – *Le modèle de mobilité.*

Pour calculer les coordonnées d'un nœud  $n$  à l'instant  $t$  durant un intervalle  $i$  d'une durée  $T_n^i$ , d'un angle  $\theta_n^i$  et d'une vitesse  $V_n^i$ , on calcule d'abord la distance  $D$  parcourue par  $n$  selon la formule  $D = V_n^i T_n^i$ . Puis, on calcule les coordonnées locales  $(x, y)$  :  $x = D \sin(\theta_n^i)$  et  $y = D \cos(\theta_n^i)$ . Enfin, on calcule les coordonnées globales par un changement de repère.

Pour obtenir un équilibre entre les arrivées et les départs dans notre région, tous les nœuds sortants de la zone de périphérie sont réinjectés dans la zone qui lui est symétriquement opposée, éliminant ainsi les effets de bord. La figure 3.18 montre la réinjection des nœuds  $M1$ ,  $M2$  et  $M3$ .

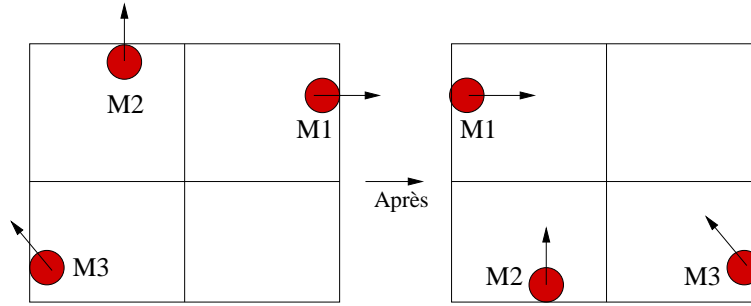


FIG. 3.18 – La réinjection des nœuds.

La mobilité d'un nœud  $n$  a besoin de trois paramètres :  $\lambda_n$ ,  $Speed\_Max$ ,  $Speed\_Min$ . Le calcul se fait comme suit : la taille des intervalles évolue selon une loi Exponentielle de paramètre  $\frac{1}{\lambda_n}$  ; la direction du mobile durant chaque intervalle suit une loi Uniforme entre  $(0, 2\pi)$  ; la vitesse du mobile durant chaque intervalle suit une loi Uniforme entre  $[Speed\_Min, Speed\_Max]$ .

### Le modèle de trafic

La GWFA génère des paquets de données UDP selon une loi Poisson de paramètre *inter-arrival*. Il représente l'intervalle de temps moyen séparant deux générations successives des paquets de données. La taille des paquets de données et la sélection d'un nœud récepteur se font selon une loi Uniforme entre  $[Taille\_min, Taille\_max]$  et  $[0, Nombre\_max\_mobiles - 1]$ . Un paquet de donnée contient trois champs : l'adresse de nœud destinataire, la taille de paquet et le numéro de séquence. Le champ numéro de séquence permet de déterminer les paquets de données perdues.

### La taille des buffers

Afin d'assurer une perte de données nulle, la taille des buffers des BSs doit être suffisamment grande pour stocker la quantité maximale des données qui peuvent être perdues durant le processus du handover. Ces paquets de données seront renvoyés par la nouvelle BS. La figure 3.19 présente le temps latence  $L_D$  nécessaire pour qu'un TM découvre qu'il doit se connecter à une nouvelle BS.

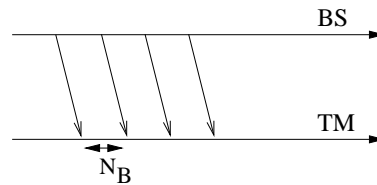


FIG. 3.19 – Le temps de réception des  $T_B$  trames balises.

La borne supérieure de  $L_D$  est donnée par  $N_B \times T_B$  où  $N_B$  est l'espace de temps entre deux trames balises et  $T_B$  présente le nombre des trames balises successives qui provoquent une initiation d'un handover en cas où elles ne sont pas reçues. Par conséquent, le nombre maximal des paquets de données qui peuvent être perdues durant un handover sans un schéma de bufferisation est de

$$(\lceil L_D \rceil / \text{inter-arrival}) + 1.$$

À partir de cette valeur, la taille maximale d'un buffer qui garantit un zéro-perte de données est extrapolée facilement. Il est clair que l'augmentation de la fréquence d'émission des trames balises réduit la taille des buffers.

### 3.10.2 Les résultats des simulations

L'objectif de ces simulations est de valider nos protocoles de gestion des handovers qui visent à minimiser la latence et la perte des paquets. Les cellules de niveau inférieur de la OVN (niveau 0) implémentent une technologie sans fil AT&T WaveLAN [40]. Cette technologie offre une bande passante de 1.6 Mb/s, 100 m de diamètre par cellule et 2 ms de temps latence. Les cellules du deuxième niveau (niveau 1) implémentent la technologie sans fil Metricom Ricochet Network [43]. Cette technologie permet à chaque cellule d'avoir une bande passante de 60 kb/s, 1km de diamètre et 100 ms de temps latence. La taille des paquets générés par la GWFA varie entre 1 kb et 65 kb. La meilleure valeur pour  $T_B$  est de 3 [44]. 80% (le seuil  $T$ ) des cellules de niveau hiérarchique inférieur sont couvertes par une cellule de niveau supérieur. Nous avons fixé le nombre des nœuds à 50 ; La vitesse maximale des nœuds est de 5 km/h.

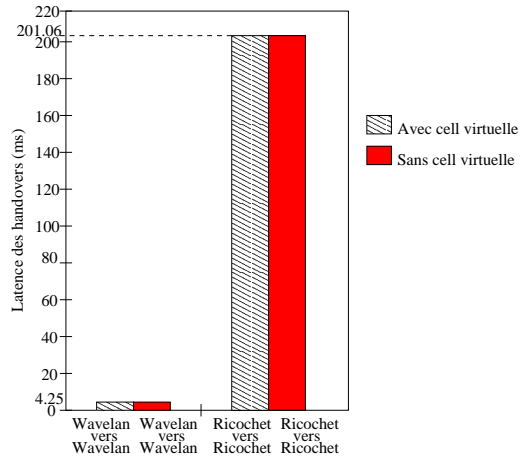


FIG. 3.20 – La latence des handovers horizontaux.

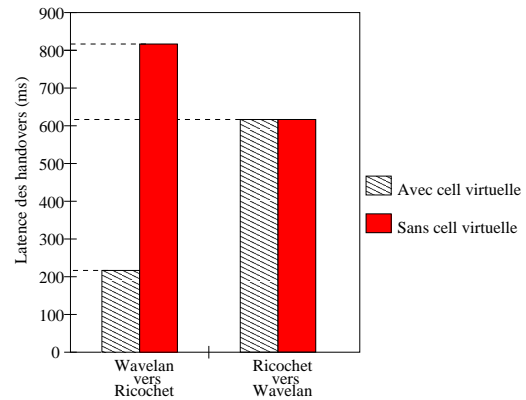


FIG. 3.21 – La latence des handovers verticaux.

La figure 3.20 montre une comparaison entre le système d'un handover horizontal sans et avec l'utilisation de la cellule virtuelle. Dans notre simulation, les handovers horizontaux possibles sont d'une cellule WaveLAN à une autre WaveLAN ou d'une cellule Ricochet à une autre Ricochet. La latence d'un handover horizontal à base de WaveLAN technologie est largement inférieure à celle d'un handover horizontal à base de Ricochet (Ricochet→Ricochet). 4,25 ms dans WaveLAN tendit que 201,6 ms dans Ricochet. Ceci est dû à la différence de la latence et la bande passante offerte par la technologie. La technologie WaveLAN a un débit plus important que celui de la technologie Ricochet et une latence très faible. Il est évident que la latence d'un système de base

d'un handover horizontal sans et avec l'utilisation de la cellule virtuelle est la même. La technique de la cellule virtuelle n'intervient que dans le cas d'un handover vertical montant.

L'efficacité des cellules virtuelles est illustrée dans la figure 3.21. Avec une période de 200 ms entre deux trames balises, les handovers verticaux montants d'une cellule WaveLAN à une cellule Ricochet réalisent une meilleure performance que les handovers verticaux descendants en termes de temps de latence. Un handover vertical montant (WaveLAN→Ricochet) consomme en moyenne 800,422 ms contre 600,285 ms pour un handover vertical descendant (Ricochet→WaveLAN), ce qui est conforme avec l'analyse présentée dans la section 3.6. L'utilisation des cellules virtuelles optimise les performances des handovers verticaux montant de 75%. La latence est passée de 800,422 ms à 200,321 ms en éliminant l'attente des 3 trames balises. Les cellules virtuelles ne peuvent pas prédire qu'un TM va recevoir  $T_B$  trames balises d'un niveau hiérarchique inférieur dans la OWN. Un TM connecté à une cellule Ricochet doit attendre impérativement la réception de 3 trames balises successives envoyée par la même cellule WaveLAN. Par conséquent, les cellules virtuelles n'influent pas sur le temps de latence des handovers verticaux descendants.

L'utilisation des cellules virtuelles s'avère très utile pour réduire le temps de latence durant les handovers. Une application classique pour mieux voir l'efficacité de cette technique est la téléphonie mobile. L'interactivité des conversations exige un délai de tolérance inférieur à une valeur maximale de 200 ms. La latence des handovers ne doit pas dépasser cette valeur maximale. Nous conservons la technologie sans fil WaveLAN de niveau hiérarchique 0 dans la OWN. Nous supposons que les cellules de niveau hiérarchique 1 dans la OWN offrent une bande passante de 1 Mb/s, une latence de 10 ms et un diamètre de 400 m. Chaque BS envoie périodiquement, chaque 100 ms, des trames balises. Afin de simuler le relaiage en temps réel des paquets de téléphonie aux TMs, la GWFA génère et envoie des paquets de données de 200 octets chaque 20 ms. Les figures 3.22 et 3.23 montrent les temps séparant la réception de deux paquets successifs. Logiquement, l'espace de temps entre la réception des paquets est environ de 20 ms. Le premier paquet reçu après un handover, paquet de numéro de séquence 13, est espacé par un écart supplémentaire de 4,2 ms qui correspond à un temps de latence d'un handover horizontal au niveau hiérarchique 0 de la OWN. Le paquet de données avec 14 comme un numéro de séquence est reçu en un délai inférieur à 20 ms ce qui est logique vu que le paquet numéro 13 est reçu en retard. La latence des handovers horizontaux est la même avec ou sans l'utilisation des cellules virtuelles. Après la réception du paquet numéro 18, le TM effectue un handover vertical montant et par conséquent le paquet numéro 19 est reçu après un temps de 322 ms dont  $3 \times 100$  ms représentant le temps de perte de 3 trames balises. Les paquets de données avec les numéros de séquence de 20 à 35 sont déjà disponibles dans le buffer de la nouvelle BS à cause du retard engendré par la transmission de paquet numéro 19. Le mécanisme de base des handovers verticaux ne permet pas l'interactivité des conversations puisque la latence introduite est supérieure à la valeur maximale autorisée. L'utilisation des cellules virtuelles réduit l'espace de temps entre le 18ème et 19ème paquet à 22 ms en éliminant la composante de latence  $T_B \times N_B$ .

D'après les numéros de séquence, aucun paquet n'est perdu ou arrivé dans le désordre. Ceci est assuré par les retransmissions des paquets après les handovers, la taille des buffers et une mobilité très faibles (2 m/s).

Lorsque les TMs se déplacent avec une vitesse très faible, le système des handovers est très efficace et assure une perte des paquets de données nulle. Avec une mobilité rapide, le nombre des handovers augmente et par conséquent, des paquets de données peuvent être perdus. Dans cette simulation, les systèmes sans fil WaveLAN et Ricochet sont considérés. L'*inter-arrival* est fixé à 60 ms et le nombre des TMs à 50. La figure 3.24 illustre une comparaison entre le système de base sans et avec la gestion de la mobilité rapide. Les courbes représentent le pourcentage de la perte des

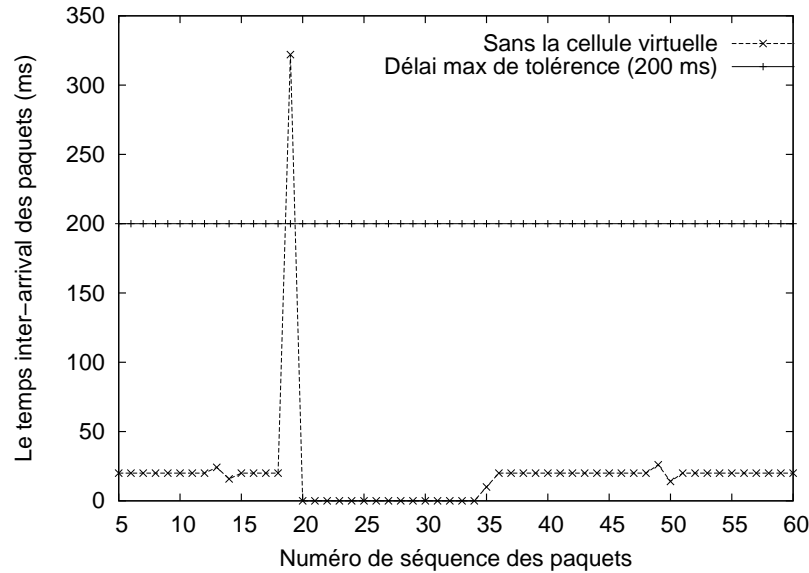


FIG. 3.22 – Le temps d'arrivée des paquets sans l'utilisation des cellules virtuelles.

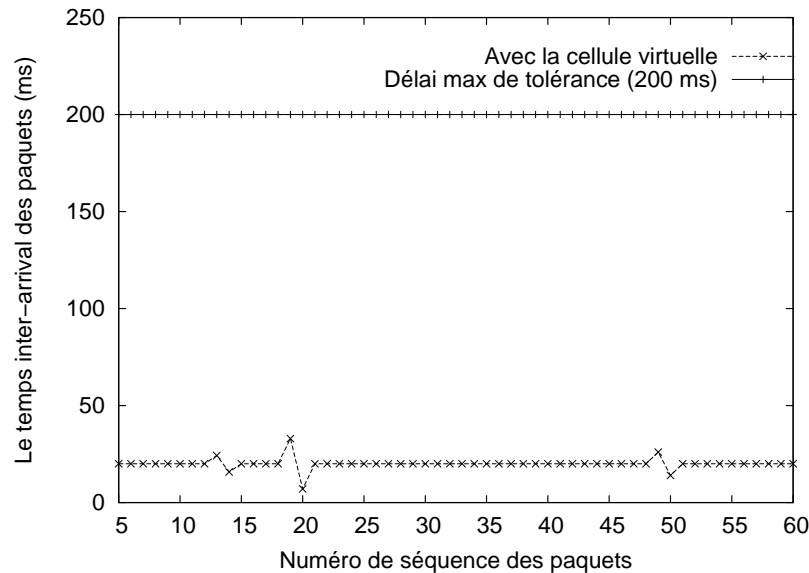


FIG. 3.23 – Le temps d'arrivée des paquets avec l'utilisation des cellules virtuelles.

paquets de données en fonction de la vitesse de mobilité. Lorsque la mobilité est inférieure à 2 m/s, une perte nulle est réalisée. Avec l'augmentation de la vitesse, les pertes de données augmentent. Ceci est due essentiellement à la taille des buffers qui ne peuvent pas stocker pendant une large durée les données à retransmettre puisque les TMs qui se déplacent à grande vitesse changent les cellules rapidement et les anciens paquets de données sont écrasés par les nouveaux. La procédure de gestion de la mobilité rapide fixe des seuils de vitesse pour chaque niveau hiérarchique de la OVN. Ce seuil doit être bien choisi pour minimiser les pertes. Comme notre OVN contient deux niveaux, seulement le seuil de niveau 0 est significatif qui permet aux TMs dont la vitesse excède

ce seuil de passer au niveau hiérarchique 1. Nous avons considéré les seuils suivants : 2 m/s (1 handover chaque 200 s), 5 m/s (1 handover chaque 40 s) et 10 m/s (1 handover chaque 20 s). Les pertes engendrées par le système doté de la procédure de gestion de mobilité rapide avec un seuil de 2 m/s présente les meilleures performances. Les pertes augmentent avec l'augmentation du seuil. 2 m/s est le seuil le plus approprié pour le niveau 0.

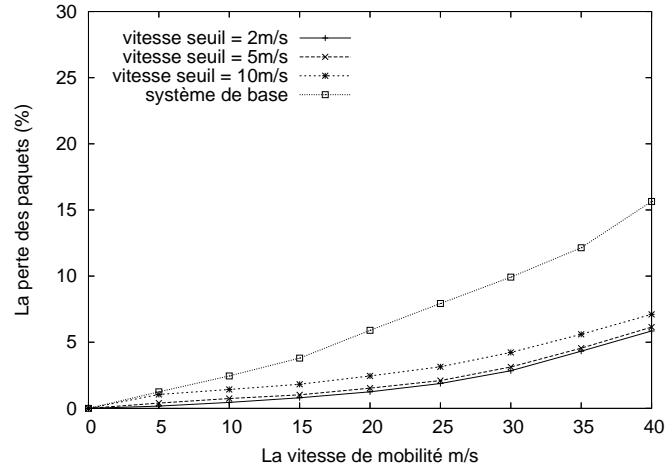


FIG. 3.24 – La perte des paquets en fonction de la mobilité.

Le tableau 3.2 montre les valeurs maximales de la latence et l'overhead engendrés par une émission rapide (100 ms) des trames balises.

Handover	L (ms)	B (bits/s)
WaveLAN→Ricochet	500,397	2480
Ricochet→WaveLAN	300,456	2480

TAB. 3.2: Émission rapide des trames balises.

L'émission rapide des trames balises permet de réduire le temps de la latence contre une consommation large de la bande passante. À partir du tableau 3.3, la latence des handovers verticaux est inférieure à celle du système en prenant  $T_D = 5$  pour le seuil des paquets (le choix de  $T_D$  est fait selon les heuristiques montrées dans [44]). Par contre, l'overhead généré est très important. L'émission rapide des trames balises présente des performances nettement supérieures à celles de diffusion multiple des paquets.

Handover	L (ms)	B (bits/s)
WaveLAN→Ricochet	500	546133
Ricochet→WaveLAN	500	546133

TAB. 3.3: Diffusion multiple des paquets.

La diffusion multiple des entêtes permet l'optimisation de l'overhead généré par les diffusions multiples des paquets (tableau 3.4).

Handover	L (ms)	B (bits/s)
WaveLAN→Ricochet	700,419	3733
Ricochet→WaveLAN	500,401	3733

TAB. 3.4: *Diffusion multiple des entêtes.*

### 3.11 Conclusion

L'utilisateur de la 4ème génération de mobiles a plusieurs technologies d'accès sans fil à sa disposition. Cet utilisateur veut pouvoir être connecté au mieux, n'importe où, n'importe quand et avec n'importe quel réseau d'accès. Pour cela, les différentes technologies sans fil doivent co-exister de manière à ce que la meilleure technologie puisse être retenue en fonction du profil de l'utilisateur et de chaque type d'application et de service qu'il demande. Nous avons proposé une structure hiérarchique des réseaux sans fil et mobiles. Cette structure est composée de plusieurs niveaux selon les caractéristiques de chaque technologie sans fil. Chaque technologie est enveloppée par une autre de large zone de couverture mais avec un débit inférieur. L'ajout et/ou la modification de certaines composante et protocoles est nécessaire afin de réaliser l'interconnexion et d'assurer un transfert cellulaire transparent entre les différentes technologies par rapport aux utilisateurs.

Dans ce contexte, l'équipement terminal devra rechercher en permanence le meilleur réseau d'accès en fonction des besoins de l'utilisateur. C'est-à-dire, le terminal mobile cherche sans cesse à descendre dans la structure hiérarchique (OWN) en exécutant des handovers verticaux descendants. L'inaccessibilité du réseau d'attachement et les niveaux inférieurs obligent le terminal mobile d'effectuer des handovers verticaux montants après un temps considérable de déconnexion. Il est nécessaire de réduire la latence des handovers verticaux montants pour assurer la continuité de service sans dégradation. Nous avons proposé deux architectures qui permettent de récupérer les paquets de données perdus durant les handovers. L'utilisation de la technique des cellules virtuelles anticipe les handovers verticaux montants et réduit au maximum la composante  $L_D$  de la latence sans signalisation supplémentaire. D'autres méthodes plus agressives ont été proposées mais seulement applicables dans le contexte d'Arch-1. L'inconvénient principal de ces techniques est l'overhead qui peut dégrader les performances lorsque plusieurs terminaux mobiles exécutent ces techniques à la fois.





## Chapitre 4

# Conception des algorithmes pour le routage à multiples contraintes

L'objectif d'un calcul de la table routage est de déterminer une route (i.e., un ensemble de liens à parcourir) respectant certaines contraintes, pour établir une connexion d'un nœud source vers un nœud destinataire. Ce calcul est inclu dans le protocole de routage qui permet la diffusion des informations nécessaires à ce calcul.

L'émergence des réseaux haut débit rend de plus en plus crucial le calcul de route entre source et destinataire. En effet :

- La qualité de service que l'on est en mesure d'offrir à une connexion est directement liée au choix du chemin. Le calcul de route doit prendre en compte les différentes contraintes imposées par la connexion (débit, délai, taux de perte, etc), ces paramètres peuvent en effet varier en fonction des liens empruntés. Il faut donc mettre en œuvre un algorithme de routage qui a pour rôle de trouver le meilleur chemin possible entre la source et le destinataire pour satisfaire les différents critères de qualité imposés.
- Le calcul doit également répartir de la manière la plus homogène possible les ressources du réseau. Il faut éviter de concentrer le trafic sur certains liens laissant d'autres inutilisés. Une répartition de la charge des ressources ne peut aboutir qu'à une amélioration des performances du réseau.
- Le calcul de route doit être le plus simple possible afin d'éviter les complexités en terme de temps d'exécution.
- Avec des réseaux qui supportent un nombre important de nœuds (les réseaux ad hoc par exemple), l'algorithme de routage doit supporter un passage à l'échelle.

Un algorithme de calcul d'une route avec QoS (Qualité de Service) consiste à trouver un chemin entre une source et une destination qui satisfait les exigences de QoS (bande passante, délai, etc) tout en utilisant d'une manière efficace les ressources du réseau (coût, équilibrer la charge, etc). Les algorithmes de calcul de la table de routage proposés dans cette thèse visent une application dans le domaine du routage point-à-point sous multiples contraintes pour assurer la gestion de la qualité de service dans les réseaux dynamiques sans aucune entité centrale. Donc le routage doit être adaptatif (dynamique) et distribué entre les nœuds du réseau. Les calculs sont faits sur un graphe topologique qui représente l'état du réseau. Ce graphe est construit par l'algorithme

de routage après des diffusions de l'information de topologie.

Les politiques de routage influent de plus en plus sur les protocoles de routage point à point déployés dans les réseaux tels qu'Internet, les réseaux ad hoc, etc. Si le calcul des routes est basé sur plusieurs contraintes, le protocole doit être capable de mesurer ces contraintes et de les diffuser dans le réseau.

## 4.1 Interconnexions et graphes

Nous présentons l'interconnexion de réseau par un graphe simple, dont les arêtes sont les liaisons, et dont les nœuds sont les équipements, stations ou routeurs. Les liaisons sont affectées d'une ou plusieurs fonctions de poids positives. Ces poids pourront représenter la distance entre les nœuds d'extrémités, le délai de transmission des données sur la liaison, le débit, le coût, etc. Nous ne prenons pas en compte le cas des liaisons redondantes ou de secours entre deux équipements de réseau ; ces liaisons seront toujours représentées par une seule arête entre deux nœuds du graphe. Nous supposons que le réseau est connexe. Toutefois, de manière transitoire, des composantes de réseau peuvent devenir momentanément isolées suite à une mobilité, panne ou autre. Le routage point-à-point dynamique réagit immédiatement en procédant à des mises à jours dans les informations de routage afin de trouver des chemins vers ces composantes. On ne considère pas l'état de réseau au moment d'une déconnexion temporaire de certains éléments mais uniquement après la réaction de protocole de routage ce qui renforce l'hypothèse que le graphe est toujours supposé connexe.

Le graphe est supposé orienté. En effet, nous considérons uniquement le cas de liaisons point à point bidirectionnelles entre deux équipements du réseau mais les valeurs des poids sur les deux arcs peuvent être différentes ; l'existence des liaisons unidirectionnelles affecte le routage point à point [45] et leur prise en compte nécessite des aménagements des protocoles de routage. Les algorithmes de construction de la table de routage que nous présentons dans ce mémoire, exploitent tout simplement les informations issues du routage point-à-point et ne nécessitent aucune hypothèse sur le type des liaisons parcourues.

Les poids des arêtes sont exploités par les algorithmes de calcul de la table de routage pour déterminer la meilleure route entre deux nœuds quelconques du graphe. Nous supposons que chaque fonction de poids est répartie d'une manière homogène sur le graphe et donne des valeurs positives dans  $\mathbb{R}^+$ .

Nous rappelons ici les notions usuelles de la théorie des graphes qui nous permettront de décrire le plus fidèlement possible les caractéristiques d'un graphe topologique.

Soit  $G = (V, E, \Omega)$  le graphe topologique pondéré, où  $V$  est l'ensemble des nœuds,  $E$  est l'ensemble des arcs et  $\Omega$  est l'ensemble des fonctions de poids associées à chaque arc. Le graphe topologique pondéré est défini par

$$\begin{aligned} E &\subset V \times V \\ \forall f \in \Omega : V \times V &\longrightarrow \mathbb{R}^+ \\ \forall u \in V, \forall v \in V, \text{ si } (u, v) \in E : f(u, v) &\text{ est finie} \end{aligned}$$

Le graphe est considéré comme orienté avec des liaisons bidirectionnelles, on aura donc :

$$\forall (u, v) \in E, \forall f \in \Omega \Rightarrow \begin{cases} (v, u) \in E \\ f(u, v) \neq f(v, u) \vee f(u, v) = f(v, u) \end{cases} \quad (4.1)$$

Nous étendons l'ensemble des fonctions de poids  $\Omega$ , défini sur  $E$ , à tout couple de nœuds du graphe :

$$\begin{aligned} \forall u \in V, \forall f \in \Omega : f(u, u) &= 0 \\ \forall (u, v) \notin E, \forall f \in \Omega : u \neq v &\iff f(u, v) = \infty \end{aligned}$$

La définition suivante sera utilisée par la suite dans les algorithmes de routage.

**Définition 4.1.1** On note le voisinage de distance  $k$  de nœud  $u$  dans un graphe  $G$  :

$$\Gamma_k(u) = \{v \in G : \text{dist}(u, v) = k\}$$

, où  $\text{dist}$  est la fonction nombre de sauts entre deux nœuds.

## 4.2 Les règles de composition des contraintes

Soit  $(G, V, E, \Omega)$  le graphe topologique pondéré avec  $|V| = n$  nœuds et  $|E| = m$  arcs. À Chaque arc  $(v_i, v_j) \in E$  est associée une ou plusieurs métriques (fonctions de poids). Soit  $f(v_i, v_j)$  l'une de ces métriques. La valeur de  $f$  sur un chemin  $p = (v_0, v_1, \dots, v_k)$  peut suivre une des règles de compositions suivantes :

- *Métrique additive* : une métrique  $f$  est additive si

$$f(p) = \sum_{i=1}^k f(v_{i-1}, v_i).$$

Il est clair que le délai (Del), la gigue (Gig), le nombre de sauts (Hop), le coût (C), etc, sont des métriques additives.

- *Métrique multiplicative* : une métrique  $f$  est multiplicative si

$$f(p) = \prod_{i=1}^k f(v_{i-1}, v_i).$$

La probabilité de succès de transmission (Stp) est une métrique multiplicative, tandis que la règle de composition de la probabilité de perte (Lp) est un peu plus complexe à qualifier. Une transformation de la probabilité de perte en une équivalente métrique multiplicative est possible de la manière suivante :

$$\begin{aligned} \text{Lp}(p) &= 1 - \text{Pst}(p) \\ &= 1 - \prod_{i=1}^k \text{Pst}(v_{i-1}, v_i) \\ &= 1 - \prod_{i=1}^k (1 - \text{Lp}(v_{i-1}, v_i)). \end{aligned}$$

- *Métrique concave* : la notion concave est très large. Mathématiquement, une fonction  $f$  est concave sur un intervalle  $I \in \mathbb{R}$  si son graphe se situe au dessus de toutes les cordes. C'est-à-dire,  $\forall (x, y) \in I^2, \forall t \in [0, 1] : f(tx + (1 - t)y) \geq t.f(x) + (1 - t).f(y)$ . Dans le contexte des graphes pondérés, une métrique  $f$  est concave si

$$f(p) = \min\{f(v_{i-1}, v_i), i = 1, \dots, k\}.$$

La bande passante (Bw) est un exemple typique de métrique concave. En fait, La bande passante du lien le moins performant qui est attribuée au chemin tout entier.

### 4.3 Résumé des algorithmes de routage existants

Dans cette section nous présentons un aperçu des différents algorithmes de routage proposés dans la littérature. Les travaux de Wang et Crowcroft [5] ont montré que le problème de trouver un chemin optimal sous la présence de deux ou plusieurs métriques additives et/ou multiplicatives est un problème NP-complet. Un problème est dit NP si, étant donné une réponse au problème, il existe un algorithme polynomial permettant de vérifier la validité de cette réponse. Un exemple classique d'un tel problème est la recherche d'un cycle hamiltonien dans un graphe (c'est-à-dire, un cycle passant par tous les sommets du graphe), il est clair qu'étant donné une solution, il est facile en  $O(n)$  de vérifier que c'est un cycle et qu'il passe par tous les sommets du graphe. Un problème ( $P$ ) est dit NP-complet s'il est dans NP, et si tout problème de type NP peut se ramener au problème ( $P$ ) à l'aide d'une transformation polynomiale. Il est conjecturé qu'il n'existe pas d'algorithme polynomial permettant de résoudre un problème NP-complet. D'autre part, seuls les algorithmes polynomiaux sont utilisables en pratique. À notre connaissance, il n'y a aucun algorithme largement admis qui peut donner la solution optimale au problème consistant à trouver un chemin sous deux contraintes additives en un temps polynomial.

Widyono [46] a proposé une méthode de routage qui donne un chemin optimal ayant le coût le plus bas possible sans violer la contrainte de délai, mais malheureusement en temps exponentiel. L'algorithme est appelé Bellman-Ford sous contraintes (CBF). CBF est un algorithme basé sur un parcours en largeur d'abord découvrant des chemins avec des délais monotones et en retenant en compte et en mettant à jour à chaque étape les nœuds visités qui ont le coût le plus bas possible. Il s'arrête, quand la contrainte la plus élevée est excédée, ou il n'y a plus de possibilité d'améliorer les chemins. Puisque cette extension utilise le délai qui est une métrique additive continue au lieu de la métrique nombre de sauts, la table de routage qui contient des entrées pour chaque destination avec son délai peut devenir très large si la méthode de codage de délai n'est pas optimisée. Comme nous l'avons mentionné, l'algorithme de CBF a un temps de fonctionnement exponentiel. Widyono n'a pas analysé ou cherché à améliorer le temps de fonctionnement de son algorithme. Il a seulement précisé sa nature exponentielle mais en même temps il a signalé que son algorithme a un temps raisonnable dans des cas réels ou en pratique.

L'algorithme CBF a subi plusieurs améliorations et extensions notamment en termes de scalabilité. Les auteurs de [47–50] ont proposé des approximations quasi-polynomiales. Ils ont prouvé que pour n'importe quelle valeur  $\varepsilon > 0$ , il existe un algorithme en temps polynomial capable de trouver un chemin satisfaisant la contrainte de délai tout en maintenant le coût sous une borne de  $1 + \varepsilon$  du coût optimal. Le temps d'exécution de la meilleure approximation est de  $O(nm \log n \log \log n + \frac{nm}{\varepsilon})$  [50]. Malheureusement, dans le cas pratique, le temps d'exécution de ces méthodes pour un  $\varepsilon$  suffisamment petit est largement supérieur à celui de la méthode CBF ce qui place ces algorithmes dans un contexte plutôt théorique.

Une méthode très simple proposée par W. C. Lee [51] qui ne donne pas des chemins optimaux, mais elle trouve une solution approximative au problème de plusieurs métriques additives. Elle consiste à calculer le plus court chemin pour la première métrique et vérifier s'il respecte les contraintes des autres métriques. Si au moins une contrainte est violée, l'algorithme reprend le processus mais avec une autre métrique et il s'arrête lorsqu'un chemin est trouvé vérifiant le reste des contraintes ou tous les plus courts chemins pour cette métrique violent au moins une des contraintes restantes. Cet algorithme est simple, il trouve rapidement une solution appropriée si elle existe, mais il n'y a aucune garantie de trouver le chemin optimal et nous ne savons rien au sujet de la qualité du chemin trouvé.

Plusieurs algorithmes comme [52, 53] proposent des extensions qui améliorent l'idée précédente de W. C. Lee. Par exemple, l'algorithme proposé par Pornavalai [54] qui consiste à calculer dans un premier temps l'ensemble des plus courts chemins pour chaque métrique de la source vers tous les nœuds du réseau et puis de tous les nœuds vers la destination. En se basant sur ces chemins, l'ensemble de toutes les combinaisons possibles est calculé et le meilleur chemin est pris en compte.

Cheng et Nahrstedt [55] ont proposé un algorithme qui trouve un chemin vérifiant les contraintes de deux métriques en un temps polynomial. L'algorithme réduit le problème original en un simple problème facile à résoudre par un algorithme de plus court chemin. La transformation du problème consiste à modifier la fonction objective (le coût par exemple) en intégrant l'autre métrique. Lorsqu'une extension de l'algorithme de Dijkstra [56] est utilisée, la complexité de calcul de leur algorithme dans un graphe de  $n$  nœuds et  $m$  arcs est de  $O(x^2n^2)$ , et de  $O(xnm)$  lorsqu'une extension de l'algorithme de Bellman-Ford [57] est appelée, où  $x$  est un nombre entier positif ajustable dont la valeur détermine la performance et le coût de l'algorithme. Afin de trouver un chemin faisable avec une grande probabilité,  $x$  doit avoir une valeur supérieure à  $10n$ . Cette approche présente une technique très simple mais malheureusement elle n'a aucune garantie de trouver une solution au problème modifié même si le problème original en possède une. Dans [58], les mêmes auteurs ont généralisé leur algorithme pour plus de deux métriques avec une complexité de calcul de  $O(x_1^2 \dots x_{k-1}^2 n^2)$  ou  $O(x_1 \dots x_{k-1} nm)$ , où  $x_1, \dots, x_{k-1}$  sont des entiers ajustables pour chaque métrique.

Les algorithmes proposés dans cette thèse sont basés sur la combinaison des principes de l'algorithme de W. C. Lee, celles de Cheng et Nahrstedt et la technique de la relaxation lagrangienne. Ils ont pour objectif de trouver d'une manière permanente des solutions optimales qui sont aussi pour le problème original ou des solutions approximatives avec un écart bien maîtrisé. Ces algorithmes sont valables pour deux, trois et quatre métriques.

Dans [59], Jaffe a présenté deux algorithmes d'approximation pour le problème de la sélection d'un chemin sous contraintes multiples. Ce problème est basé sur la minimisation de la fonction  $f(p) + d.g(p)$ , où la valeur initiale de  $d$  est égale à 1 dans la première approximation et  $\sqrt{\Delta_f/\Delta_g}$  dans la deuxième approximation,  $g$  et  $f$  sont deux métriques additives. Jaffe a prouvé que la deuxième approximation présente de meilleures performances que la première.

Dans [60], les auteurs ont utilisé les  $k$  plus courts chemins avec une fonction coût non linéaire pour résoudre un problème de plus de deux contraintes. L'algorithme résultant a une complexité de calcul de  $O(km \log(kn) + k^3m)$ , où  $k$  présente le nombre des plus courts chemins. La performance de cet algorithme dépend fortement de la valeur de  $k$ . Si  $k$  prend des larges valeurs, l'algorithme trouve des chemins de performances proches à celles des chemins optimaux, mais le temps de calcul diverge rapidement et il devient exponentiel.

Récemment, Une simple et efficace heuristique pour trouver un chemin en fonction de deux métriques  $w_1$  et  $w_2$  a été proposé dans [61]. Cet algorithme consiste à minimiser la fonction  $\alpha w_1(e) + \beta w_2(e)$ , où  $\alpha, \beta \in \mathbb{Z}^+$  en utilisant Dijkstra comme un algorithme de plus court chemin.

La complexité du calcul de cet algorithme et de  $\log W$  multiplié par la complexité de l'algorithme de Dijkstra, où  $W$  est la borne supérieure en terme de coût du plus long chemin qui satisfait une des deux métriques.

#### 4.4 Routage à contrainte simple

Dans le routage classique, les routes sont calculées selon une métrique simple comme le nombre de sauts ou le délai en utilisant des algorithmes de plus court chemin. Le routage avec QoS est plus complexe car les décisions de routage se basent sur  $n$  métriques.

Pour une simple métrique additive  $f$ , le problème de routage consiste à trouver un chemin  $p^*$  entre un nœud source  $s$  et un nœud destinataire  $d$  tel que la valeur  $f(p^*)$  est la plus faible parmi toutes les valeurs de tous les chemins entre  $s$  et  $d$ . Dans ce cas, un algorithme de plus court chemin comme celui de Dijkstra [56] ou Bellman-Ford [57] est employé. Il construit non seulement une route de  $s$  vers  $d$  mais un arbre de recouvrement minimal de racine  $s$ .

Pour une simple métrique multiplicative comme la probabilité de perte, le problème de routage consiste à trouver un chemin  $p^*$  de valeur  $f(p^*)$  minimale entre la source  $s$  et la destination  $d$ . Dans ce cas, une variante des algorithmes de plus court chemin est utilisée. Elle consiste à modifier l'addition dans le processus original de la relaxation des arcs par la multiplication. La probabilité de perte est la métrique la plus utilisée dans ce genre de routage.

Le problème de routage avec une métrique concave  $f$  consiste à trouver un chemin  $p^*$  entre  $s$  et  $d$  qui maximise  $f(p^*)$ . Ce calcul de route fait appel à une variante des algorithmes de plus court chemin où la relaxation des arcs est basée sur la valeur minimale des poids.

La complexité de l'algorithme de Dijkstra ou ses variantes dans un graphe avec des arcs de valeur non négative dépend du choix de la structure de donnée des étiquettes : si c'est une liste, elle est de  $O(|E| |V| + |V|^2)$  [56] ; si c'est un arbre binaire équilibré, elle est de  $O(|E| \log |V| + |V| \log |V|)$  [56]. On peut aussi utiliser une pile de Fibonacci [62] pour laquelle l'extraction du minimum s'effectue en temps  $\log$ , et l'insertion en temps amorti  $O(1)$ , on obtient alors une complexité en  $O(|V| \log |V| + |E|)$ . Tandis que la complexité de l'algorithme de Bellman-Ford est de  $O(|E| |V|)$  [57]. L'algorithme fonctionne même si certains poids des arcs sont négatifs. Il détecte aussi l'existence d'un circuit de poids négatif.

#### 4.5 Routage aux contraintes multiples

Wang et Crowcroft [5] ont démontré que le problème de trouver un chemin dans un graphe topologique sous  $n$  métriques additives et  $m$  métriques multiplicatives est NP-complet si  $n + m \geq 2$ . En plus, avec plusieurs métriques, le meilleur chemin qui garantit l'optimalité de toutes les métriques à la fois peut ne pas exister.

L'hypothèse sur la corrélation entre les métriques est un élément déterminant dans le processus de résolution du problème. Si la corrélation est très forte entre les métriques, et c'est le cas dans la pratique, l'algorithme de résolution se complique au niveau de la détermination de quelle façon les métriques sont reliées. Le délai de transmission de bout en bout dépend de la capacité des liens, le délai de propagation, le nombre de sauts, la probabilité de perte sur les liens et le modèle de trafic employé. Une fois la corrélation est explicite, la recherche d'un chemin devient une tâche facile. Ma et Steenkiste [63] ont proposé des algorithmes basés sur des métriques dépendantes entre elles et ils ont précisé que toute la difficulté réside dans la détermination de la relation entre

les métriques. Si la corrélation est très faible, les métriques sont considérées indépendantes entre elles et la complexité du calcul devient le principal défi.

#### 4.5.1 Une approche à métrique unique mixte

Une approche évidente consiste à exprimer les besoins de QoS de l'utilisateur en une mesure unique (un coût total) et l'utiliser ensuite comme une seule métrique. Par exemple, on peut exprimer les métriques de la bande passante (Bw), le délai (Del) et la probabilité de perte (Lp) sur un chemin  $p$  par une fonction  $f$  proportionnelle à la bande passante, et inversement proportionnelle au délai et à la probabilité de perte :

$$f(p) = \frac{Bw(p)}{Del(p) \times Lp(p)}.$$

L'utilisation de la fonction coût total présente deux problèmes : i) maximiser  $f(p)$  ne vérifie pas forcément les besoins de QoS de chaque métrique, ii) la règle de composition de cette fonction n'est pas claire car les règles des différents paramètres sont distinctes. Malgré ces problèmes, la métrique unique mixte est utilisée dans plusieurs approches [64, 65]. Comme nous l'avons mentionné la métrique unique mixte ne garantit pas les contraintes de QoS individuellement, mais les poids des différentes métriques peuvent être ajustés dans le calcul de la fonction coût total pour augmenter l'importance d'un besoin de QoS spécifique.

Soit  $s$  le nœud source,  $S$  l'ensemble des nœuds où les meilleurs chemins vers ces nœuds sont déjà déterminés.  $V - S$  le reste des nœuds. Soit "d" un tableau des meilleures valeurs estimées de la métrique unique mixte pour chaque nœud et "prédécesseur" le tableau des nœuds prédécesseurs. L'algorithme modifié de Dijkstra pour la métrique unique mixte est le suivant :

**Étape 1 :**  $S = \emptyset$ ,  $d[s] = \max$ ,  $\text{prédécesseur}[k] = -1$ ,  $d[k] = 0$ ,  
pour chaque  $k \neq s$ ,  $k \in V - S$  ;

**Étape 2 :** Extraire le nœud  $u \in V - S$  qui a une valeur  $d[u]$  maximale ;

**Étape 3 :**  $S = S \cup \{u\}$  ;

**Étape 4 :** Pour chaque nœud  $v \in \Gamma_1(u)$  :

$$\begin{aligned} \text{Si } d[v] &< \frac{\min\{Bw(u), Bw_{uv}\}}{(Del(u) + Del_{uv}) \times (1 - (Pst(u) \times Pst_{uv}))} \text{ alors} \\ d[v] &= \frac{\min\{Bw(u), Bw_{uv}\}}{(Del(u) + Del_{uv}) \times (1 - (pst(u) \times pst_{uv}))} ; \text{prédécesseur}[v] = u ; \end{aligned}$$

**Étape 5 :** Si  $V - S = \emptyset$  alors c'est terminé. Sinon, aller à l'étape 2 ;

L'étape 4 met à jour les meilleures estimations de chaque nœud  $v \in V$  connecté au nœud  $u$ . Cet algorithme a la même complexité que l'algorithme classique de Dijkstra.

#### 4.5.2 Une approche à plusieurs métriques

On a vu précédemment que la présence de deux ou plusieurs métriques additives et/ou multiplicatives rend le problème NP-complet. Dans ce cas, les solutions ne sont pas forcément optimales et des heuristiques qui donnent des solutions proches de l'optimalité avec des écarts maîtrisables sont nécessaires. Comme nous l'avons précisé, un algorithme de routage avec QoS doit trouver un chemin respectant les contraintes de QoS tout en minimisant les ressources du réseau. Dans



ce contexte, même si les applications exigent des contraintes sur la bande passante ou le délai par exemple, nos algorithmes rajoutent un aspect de minimisation de certaines ressources comme le coût ou le nombre de sauts.

### Une métrique concave et une métrique additive

Nous prenons la bande passante (Bw) comme une métrique concave et le délai (Del) comme un exemple d'une métrique additive. Un chemin entre un nœud source  $s$  et un nœud destinataire  $d$  garantissant une bande passante maximale et un délai minimum peut ne pas exister. Donc, il n'est pas toujours possible de satisfaire les deux métriques à la fois. Définir une priorité ou une hiérarchie en terme de dominance afin de déterminer le meilleur chemin devient une nécessité.

Le délai a trois composantes : le temps de séjour dans la file d'attente (le temps de stockage), le temps de transmission et le temps de propagation. Le temps de séjour dans la file d'attente est dynamique selon le trafic. La bande passante est une métrique déterminante et critique pour les applications multimedia. Si la bande passante est insuffisante, la probabilité de perte et le délai de stockage sont très importants. Le temps de transmission dépend de la taille des paquets et du débit, tandis que le temps de propagation est fixe selon le support de transmission utilisé. Par conséquence, le délai dépend de la bande passante et donc il a moins de priorité que cette dernière. Notre stratégie consiste à trouver un chemin de meilleure bande passante (le plus large chemin), et lorsque les chemins les plus larges sont multiples, on choisit parmi eux celui qui donne un délai minimum (le plus court chemin). Un tel chemin est appelé le chemin le plus court parmi les plus larges. Le chemin  $p^*$  le plus large entre deux nœuds  $s$  et  $d$  est le chemin qui maximise  $Bw(p^*)$ . Un algorithme modifié basé sur les algorithmes de plus court chemin peut être utilisé. Les inconvénients de ces algorithmes se résument dans la présence de plusieurs solutions (les chemins les plus larges) de même bande passante et la possibilité que ces chemins possèdent des circuits. Un algorithme de shortest-widest chemin permet d'éviter ces inconvénients en calculant le plus court chemin parmi les chemins les plus larges (shortest-widest chemin).

**Théorème 4.5.1** *Dans un calcul distribué, un shortest-widest chemin est sans circuits.*

PREUVE :

La démonstration du théorème 4.5.1 peut se faire par contradiction. Supposons que les nœuds  $A$  et  $B$  sont impliqués dans un circuit pour une destination  $C$  (figure 4.1).

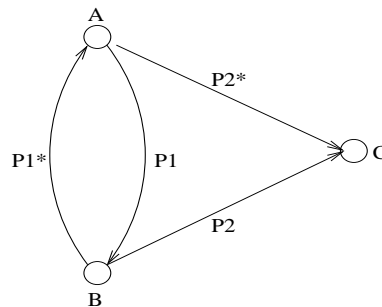


FIG. 4.1 –  $A$  et  $B$  impliqués dans un circuit.

Soit  $p_1 p_2$  un shortest-widest chemin de  $A$  à  $C$  et  $p_1^* p_2^*$  un shortest-widest chemin de  $B$  à  $C$ .

Par la définition de shortest-widest chemin, on a :

$$\text{Bw}(p_2^*) \leq \text{Bw}(p_1 p_2) \quad (4.2)$$

$$\text{Bw}(p_2) \leq \text{Bw}(p_1^* p_2^*) \quad (4.3)$$

Noter que

$$\text{Bw}(p_1^* p_2^*) = \min[\text{Bw}(p_1^*), \text{Bw}(p_2^*)] \leq \text{Bw}(p_2^*) \quad (4.4)$$

D'une manière similaire,

$$\text{Bw}(p_1 p_2) = \min[\text{Bw}(p_1), \text{Bw}(p_2)] \leq \text{Bw}(p_2) \quad (4.5)$$

À partir de ( 4.2), ( 4.4) et ( 4.5), on a

$$\text{Bw}(p_1^* p_2^*) \leq \text{Bw}(p_2) \quad (4.6)$$

Comparons (4.6) avec (4.3), on a

$$\text{Bw}(p_1^* p_2^*) = \text{Bw}(p_2) \quad (4.7)$$

D'une manière similaire, On a

$$\text{Bw}(p_1 p_2) = \text{Bw}(p_2^*) \quad (4.8)$$

L'équation ( 4.7) montre que les chemins  $p_1^* p_2^*$  et  $p_2$  ont la même bande passante. Par supposition, le chemin  $p_1^* p_2^*$  est un shortest-widest chemin, on a

$$\text{Del}(p_2) \geq \text{Del}(p_1^* p_2^*) > \text{Del}(p_2^*) \quad (4.9)$$

L'équation ( 4.8) montre que les chemins  $p_1 p_2$  et  $p_2^*$  ont la même bande passante. Par supposition, le chemin  $p_1 p_2$  est le plus court des larges, on a

$$\text{Del}(p_2^*) \geq \text{Del}(p_1 p_2) > \text{Del}(p_2) \quad (4.10)$$

Les équations ( 4.9) et ( 4.10) sont en contradiction (mutuelle).

◆

Soit  $(G, V, E, \Omega)$  le graphe topologique pondéré, où  $\Omega = \{\text{Bw}, \text{Del}\}$ . Soient “d1” et “d2” deux tableaux des meilleures estimations de la bande passante et le délai respectivement jusqu'au chaque nœud. L'algorithme de shortset-widest chemin est le suivant :

- Étape 1 :**  $S = \emptyset$ ,  $d1[s] = \max$ ,  $d1[s] = 0$ , prédécesseur $[k] = -1$ ,  
 $d1[k] = 0$ ,  $d2[k] = \infty$ , pour chaque  $k \neq s$ ,  $k \in V - S$ ;
- Étape 2 :** Extraire le nœud  $u \in V - S$  qui a une valeur  $d1[u]$  maximale ;  
 en cas d'égalité, extrait celui qui a une valeur  $d1[u]$  minimale
- Étape 3 :**  $S = S \cup \{u\}$  ;
- Étape 4 :** Pour chaque nœud  $v \in \Gamma_1(u)$  :  
 Si  $d1[v] < \min\{d1[v], Bw(u, v)\}$  alors  
 $d1[v] = \min\{d1[v], Bw(u, v)\}$  ;  $d2[v] = d2[v] + Del(u, v)$  ;  
 prédécesseur $[v] = u$  ;
- Étape 5 :** Si  $V - S = \emptyset$  alors c'est terminé.  
 Sinon, aller à l'étape 2 ;

L'étape 2 permet d'augmenter une branche de l'arbre de recouvrement par un nœud  $u$  de telle façon que la branche entière représente le plus court chemin parmi les plus larges de  $s$  à  $u$ . Cet algorithme a la même complexité que celui de Dijkstra.

### Deux métriques additives

Il est clair que le problème de trouver le meilleur chemin sous deux contraintes additives est un problème NP-complet [5]. Il n'existe pas d'algorithme polynomial qui trouve toujours la solution optimale. Nous proposons une heuristique quasi-polynomiale qui vise à minimiser une métrique additive tandis que l'autre métrique reste sous une borne supérieure. Par exemple, trouver un chemin minimal en termes de nombre de sauts avec un délai inférieur à une valeur donnée (DCLH : Delay Constrained Least Hop-Count path problem), ou bien minimiser le coût sous la contrainte de délai (DCLC : Delay Constrained Least Cost path problem), etc.

À Chaque arc du graphe topologique est associé deux métriques additives **met1** et **met2**. La description formelle pour minimiser met1 en satisfaisant la contrainte de met2 est :

$$\begin{array}{ll}
 \text{Minimiser} & \text{met1}(p) \\
 \text{sous les contraintes} & \text{met2}(p) \leq \Delta_{\text{met2}} \\
 & p \in P(s, t),
 \end{array} \tag{4.11}$$

où  $P(s, t)$  est l'ensemble des chemins entre la source  $s$  et la destination  $t$ . Notre heuristique est basée sur des techniques dites **la relaxation**. Ces méthodes cherchent à relâcher une ou plusieurs contraintes de façon à construire un nouveau problème plus simple à résoudre que le problème initial. L'avantage de ces méthodes réside dans le fait que la valeur optimale du nouveau problème est une borne pour l'optimum du problème initial. La classe de relaxation qui nous intéresse est **la relaxation lagrangienne**. Pour les problèmes dits "difficiles", c'est le cas des problèmes NP-complets, l'objectif de la relaxation lagrangienne est de relâcher les contraintes qui rendent ce problème "difficile" de façon à construire un problème admettant un algorithme polynomial. [66] constitue le premier travail de synthèse sur cette méthode. On trouvera des compléments intéressants dans [67, 68].

**La relaxation lagrangienne basée sur deux métriques additives (RLMA) :** La relaxation lagrangienne permet de négliger certaines contraintes et de les rajouter dans la fonction objectif. On considère le problème suivant :

$$\begin{array}{ll} \text{Minimiser} & f(x) \\ \text{sous les contraintes} & g(x) \leq 0 \\ & x \in X \subseteq \mathfrak{R}^n \end{array} \quad (4.12)$$

Lorsque la contrainte  $g(x) \leq 0$  est relâchée, un vecteur de multiplicateurs de lagrange  $\lambda = (\lambda_1, \dots, \lambda_m)^t \in \mathfrak{R}^m$  sera introduit. Le nouveau problème de la relaxation lagrangienne est

$$\begin{array}{ll} \text{Minimiser} & f(x) + \lambda^t g(x) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) \\ \text{sous les contraintes} & x \in X \subseteq \mathfrak{R}^n \end{array} \quad (4.13)$$

Les multiplicateurs de Lagrange sont des paramètres et non pas des variables. La fonction objective modifiée  $f_\lambda(x) := f(x) + \lambda^t g(x) = f(x) + \sum \lambda_i g_i(x)$  est appelée la fonction de lagrange du problème 4.12.

**Lemme 4.5.1** Si  $\lambda \geq 0$ , alors le problème 4.13 est une relaxation du problème 4.12.

PREUVE :

Voir [69]



**Lemme 4.5.2** Supposons que  $\bar{x}_\lambda$  est la solution optimale du problème 4.13. Alors  $\bar{x}_\lambda$  est une solution optimale du problème 4.12 si

1.  $g(\bar{x}_\lambda) \leq 0$
2.  $\lambda_i g_i(\bar{x}_\lambda) = 0, i = 1, \dots, m$
3.  $\lambda \geq 0$ .

PREUVE :

Voir [69]



Reprenons notre problème 4.11 à deux métriques additives. Afin que le problème 4.11 soit de la même forme que le problème 4.12, il faut déplacer  $\Delta_{\text{met2}}$  vers la gauche de l'interpolation. On aura donc le problème suivant :

$$\begin{array}{ll} \text{Minimiser} & \text{met1}(p) \\ \text{sous les contraintes} & \text{met2}(p) - \Delta_{\text{met2}} \leq 0 \\ & p \in P(s, t), \end{array} \quad (4.14)$$

Maintenant, la relaxation du problème 4.14 est

$$\begin{array}{ll} \text{Minimiser} & \text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}}) \\ \text{sous les contraintes} & p \in P(s, t), \end{array} \quad (4.15)$$

Pour une valeur fixe de  $\lambda$ , on peut facilement calculer un chemin optimal  $p_\lambda$  en utilisant la fonction de Lagrange  $\text{met1}_\lambda = \text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}})$ . Si  $\lambda = 0$  et  $\text{met2}(p) \leq \Delta_{\text{met2}}$  alors la solution trouvée est optimale pour le problème original. Si  $\text{met2}(p_\lambda) > \Delta_{\text{met2}}$ , la valeur de  $\lambda$  est augmentée, dans le but d'augmenter la dominance de met2 dans la fonction objective modifiée. Avec l'aide de la relaxation lagrangienne, nous avons un algorithme qui peut trouver la valeur optimale de  $\lambda$  pour une paire de source et destination. Pour trouver la meilleure valeur de  $\lambda$  qui donne la meilleure solution, on utilise les lemmes suivants.

**Lemme 4.5.3** Soit  $L(\lambda) = \min\{\text{met1}_\lambda(p) : p \in P(s, t)\}$ . Alors  $\forall \lambda \geq 0$ ,  $L(\lambda)$  est la borne inférieure du problème 4.14.

PREUVE :

Soient  $p^*$  la solution optimale (le chemin optimal) du problème 4.14. Alors

$$\begin{aligned} L(\lambda) &= \min\{\text{met1}_\lambda(p) : p \in P(s, t)\} \\ &\leq \text{met1}_\lambda(p^*) \\ &\leq \text{met1}(p^*) + \lambda(\text{met2}(p^*) - \Delta_{\text{met2}}) \\ &\leq \text{met1}(p^*). \end{aligned}$$

Afin d'obtenir la meilleure borne inférieure, on doit maximiser la fonction  $L(\lambda)$ , i.e., trouver  $L(\lambda^*) := \max_{\lambda \geq 0} L(\lambda)$ .

◆

D'après le lemme 4.5.3, la meilleure valeur de  $\lambda$  est celle qui maximise la fonction  $L$ .

**Lemme 4.5.4** La fonction  $L$  est concave et linéaire pour tout  $\lambda \geq 0$ .

PREUVE :

Soit  $t \in [0, 1]$  et  $\lambda \geq 0$ ,

$$\begin{aligned}
L(t\lambda + (1-t)\lambda) &= \min\{\text{met1}_{t\lambda+(1-t)\lambda}(p) : p \in P(s,t)\} \\
&= \min\{\text{met1}(p) + (t\lambda + (1-t)\lambda)(\text{met2}(p) - \Delta_{\text{met2}}) : p \in P(s,t)\} \\
&= \min\{\text{met1}(p) + t\lambda(\text{met2}(p) - \Delta_{\text{met2}}) + \\
&\quad (1-t)\lambda(\text{met2}(p) - \Delta_{\text{met2}}) : p \in P(s,t)\} \\
&= \min\{t\text{met1}(p) + (1-t)\text{met1}(p) + t\lambda(\text{met2}(p) - \Delta_{\text{met2}}) + \\
&\quad (1-t)\lambda(\text{met2}(p) - \Delta_{\text{met2}}) : p \in P(s,t)\} \\
&= \min\{t(\text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}})) + \\
&\quad (1-t)(\text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}})) : p \in P(s,t)\} \\
&\geq t.\min\{\text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}})\} + \\
&\quad (1-t).\min\{\text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}}) : p \in P(s,t)\}.
\end{aligned}$$

et donc la fonction  $L$  est concave. Il est évident que le fonction  $L$  est linéaire.

◆

**Lemme 4.5.5** Pour toute valeur  $\lambda \geq 0$  et un chemin  $p_\lambda$  optimal (qui a la valeur minimale) en termes de  $\text{met1}_\lambda$ , i.e.,  $p_\lambda$  est  $\text{met1}_\lambda$ -minimal, alors  $\text{met2}(p_\lambda) - \Delta_{\text{met2}}$  est le sous gradient de la fonction  $L$  au point  $\lambda$ .

PREUVE :

Soit  $\lambda_0 \geq 0$  et  $p_{\lambda_0}$  le chemin optimal en terme de  $\text{met1}_\lambda$ , i.e.,  $\min\{\text{met1}_{\lambda_0}(p) : p \in P(s,t)\} = \text{met1}_{\lambda_0}(p_{\lambda_0})$ . La valeur scalaire  $\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}}$  est le sous-gradient de la fonction  $L$  au point  $\lambda_0$  ssi (voir figure 4.2)

$$\forall \lambda \geq 0, L(\lambda) \leq L(\lambda_0) + \text{met2}(p_{\lambda_0}).(\lambda - \lambda_0).$$

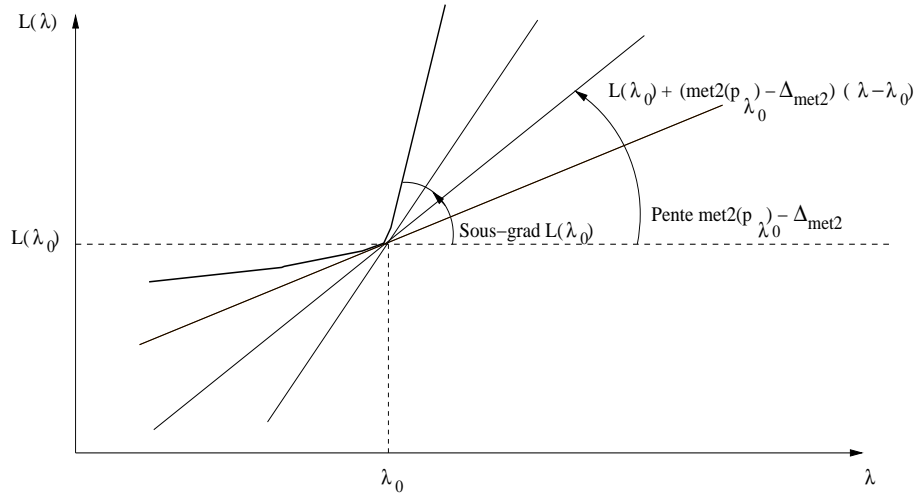


FIG. 4.2 – Sous-gradient.

Rappelons l'expression de la fonction  $L$

$$\begin{aligned}
L(\lambda) &= \min\{\text{met1}_\lambda(p) : p \in P(s,t)\} \\
&= \min\{\text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}}) : p \in P(s,t)\}
\end{aligned}$$

Par définition de  $L(\lambda)$ , on a alors

$$\forall \lambda \geq 0, \forall p \in P(s,t), L(\lambda) = \text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}}).$$

En particulier, pour  $P_{\lambda_0}$

$$\forall \lambda \geq 0, L(\lambda) \leq \text{met1}(P_{\lambda_0}) + \lambda(\text{met2}(P_{\lambda_0}) - \Delta_{\text{met2}}). \quad (4.16)$$

Or, par définition de  $P_{\lambda_0}$

$$\begin{aligned}
L(\lambda_0) &= \text{met1}_{\lambda_0}(p_{\lambda_0}) \\
&= \text{met1}(p_{\lambda_0}) + \lambda_0(\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}}).
\end{aligned} \quad (4.17)$$

Enfin, en retranchant (4.17) à (4.16), on obtient

$$\forall \lambda \geq 0, L(\lambda) - L(\lambda_0) \leq (\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}})(\lambda - \lambda_0). \quad (4.18)$$

Par conséquent  $(\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}})$  est un sous-gradient de  $L$  au point  $\lambda_0$ .

◆

**Lemme 4.5.6** *Pour n'importe quel chemin  $p_\lambda$  optimal en terme de  $\text{met1}_\lambda$  ( $p_\lambda$  est  $\text{met1}_\lambda$ -minimal), si  $\lambda < \lambda^*$  alors  $\text{met2}(p_\lambda) \geq \Delta_{\text{met2}}$  et si  $\lambda > \lambda^*$  alors  $\text{met2}(p_\lambda) \leq \Delta_{\text{met2}}$ .*

PREUVE :

Puisque  $\lambda^*$  est la meilleure valeur qui maximise la fonction  $L$ , alors on a

$$\forall \lambda \geq 0, \lambda \neq \lambda^*, L(\lambda^*) - L(\lambda) \geq 0. \quad (4.19)$$

D'après le lemme 4.5.5, la valeur scalaire  $\text{met2}(p_\lambda) - \Delta_{\text{met2}}$  est le sous gradient de la fonction  $L$  au point  $\lambda$ , i.e.,

$$\forall \lambda_1 \geq 0, L(\lambda_1) - L(\lambda_0) \leq (\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}})(\lambda_1 - \lambda_0).$$

En particulier, pour la valeur  $\lambda^*$

$$L(\lambda^*) - L(\lambda_0) \leq (\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}})(\lambda^* - \lambda_0). \quad (4.20)$$

D'après (4.19) et (4.20), on a

$$(\text{met2}(p_{\lambda_0}) - \Delta_{\text{met2}})(\lambda^* - \lambda_0) \geq 0. \quad (4.21)$$

Par conséquent, si  $\lambda < \lambda^*$  alors  $\text{met2}(p_\lambda) \geq \Delta_{\text{met2}}$  et si  $\lambda > \lambda^*$  alors  $\text{met2}(p_\lambda) \leq \Delta_{\text{met2}}$ .



**Lemme 4.5.7** *La fonction  $L$  est maximisée par une valeur ssi il existe deux chemins  $p_{met1}$  et  $p_{met2}$  qui sont  $met1_{\lambda^*}$ -minimaux et  $met2(p_{met1}) \geq \Delta_{met2}$  et  $met2(p_{met2}) \leq \Delta_{met2}$ . Si  $p_{met1}$  et  $p_{met2}$  présentent le même chemin, alors  $met2(p_{met2}) = met2(p_{met1}) = \Delta_{met2}$ .*

PREUVE :

Ce lemme est une conséquence directe du lemme 4.5.6. Plus formellement : Soit  $\lambda^*$  la valeur qui maximise la fonction  $L$ , et  $\lambda_0$  la valeur pour laquelle il existe deux chemins  $p_{met1}$  et  $p_{met2}$  qui sont  $met1_{\lambda_0}$ -minimaux. On cherche à démontrer que  $L(\lambda^*) = L(\lambda_0)$ .

D'après le lemme 4.5.1 et 4.5.5, on a

$$\begin{cases} 0 \leq L(\lambda^*) - L(\lambda_0) \leq (met2(p_{met1}) - \Delta_{met2})(\lambda^* - \lambda_0) \\ 0 \leq L(\lambda^*) - L(\lambda_0) \leq (met2(p_{met2}) - \Delta_{met2})(\lambda^* - \lambda_0) \end{cases} \quad (4.22)$$

On distingue 3 cas possibles :  $\lambda_0 < \lambda^*$ ,  $\lambda_0 = \lambda^*$  et  $\lambda_0 > \lambda^*$ .

- Si  $\lambda_0 < \lambda^*$ , et par le fait que  $met2(p_{met2}) \leq \Delta_{met2}$ , la deuxième formule dans 4.22 donne  $0 \leq L(\lambda^*) - L(\lambda_0) \leq 0$  et donc  $L(\lambda^*) = L(\lambda_0)$ .
- Si  $\lambda_0 = \lambda^*$ , et les deux formules dans 4.22 donnent  $0 \leq L(\lambda^*) - L(\lambda_0) = 0$  et donc  $L(\lambda^*) = L(\lambda_0)$ .
- Si  $\lambda_0 > \lambda^*$ , et par le fait que  $met2(p_{met1}) \geq \Delta_{met2}$ , la première formule dans 4.22 donne  $0 \leq L(\lambda^*) - L(\lambda_0) \leq 0$  et donc  $L(\lambda^*) = L(\lambda_0)$ .



**Lemme 4.5.8** *Soient  $0 \leq \lambda_1 < \lambda_2$ , et  $p_{\lambda_1}, p_{\lambda_2} \in P(s, t)$   $\lambda_1$ -minimal et  $\lambda_2$ -minimal respectivement. Alors  $met1(p_{\lambda_1}) \leq met1(p_{\lambda_2})$  et  $met2(p_{\lambda_1}) \geq met2(p_{\lambda_2})$ .*

PREUVE :

Soient  $0 \leq \lambda_1 < \lambda_2$ , et  $p_{\lambda_1}, p_{\lambda_2} \in P(s, t)$   $\lambda_1$ -minimal et  $\lambda_2$ -minimal respectivement.

D'après le lemme 4.5.5, on a

$$\forall \lambda \geq 0, \quad L(\lambda) - L(\lambda_1) \leq (met2(p_{\lambda_1}) - \Delta_{met2})(\lambda - \lambda_1).$$

En particulier, pour la valeur  $\lambda_2$

$$L(\lambda_2) - L(\lambda_1) \leq (met2(p_{\lambda_1}) - \Delta_{met2})(\lambda_2 - \lambda_1). \quad (4.23)$$

D'une manière similaire, on obtient

$$L(\lambda_1) - L(\lambda_2) \leq (met2(p_{\lambda_2}) - \Delta_{met2})(\lambda_1 - \lambda_2). \quad (4.24)$$

En additionnant (4.23) à (4.24), on aura



$$0 \leq (\text{met2}(p_{\lambda_1}) - \text{met2}(p_{\lambda_2}))(\lambda_2 - \lambda_1). \quad (4.25)$$

Puisque  $0 \leq \lambda_1 < \lambda_2$ , la formule (4.25) implique que  $\text{met2}(p_{\lambda_1}) \geq \text{met2}(p_{\lambda_2})$ .  
Maintenant il reste à démontrer que  $\text{met1}(p_{\lambda_1}) \leq \text{met1}(p_{\lambda_2})$ . Par définition de  $P_{\lambda_0}$

$$\begin{aligned} L(\lambda_1) &= \text{met1}_{\lambda_1}(p_{\lambda_1}) \\ &= \text{met1}(p_{\lambda_1}) + \lambda_1(\text{met2}(p_{\lambda_1}) - \Delta_{\text{met2}}). \end{aligned} \quad (4.26)$$

Par définition de  $L(\lambda)$ , on a alors

$$\forall \lambda \geq 0, \forall p \in P(s, t), \quad L(\lambda) = \text{met1}(p) + \lambda(\text{met2}(p) - \Delta_{\text{met2}}).$$

Pour  $\lambda = \lambda_1$ ,

$$\forall p \in P(s, t), \quad L(\lambda_1) = \text{met1}(p) + \lambda_1(\text{met2}(p) - \Delta_{\text{met2}}).$$

En particulier, pour  $P_{\lambda_2}$

$$L(\lambda_1) \leq \text{met1}(P_{\lambda_2}) + \lambda_1(\text{met2}(P_{\lambda_2}) - \Delta_{\text{met2}}). \quad (4.27)$$

Enfin, en retranchant (4.26) à (4.27), on obtient

$$\text{met1}(p_{\lambda_1}) - \text{met1}(p_{\lambda_2}) \leq 0. \quad (4.28)$$

◆

À partir des deux derniers lemmes, On peut calculer facilement  $\lambda^*$  qui maximise la fonction  $L(\lambda)$  afin d'obtenir la meilleure valeur de la fonction objectif.  $\lambda^*$  est la plus petite valeur pour laquelle il existe un chemin  $p_\lambda$  qui est  $\text{met1}_{\lambda^*}$ -minimal et vérifiant  $\text{met2}(p_\lambda) \leq \Delta_{\text{met2}}$ . Puisque  $p_{\text{met1}}$  et  $p_{\text{met2}}$  sont  $\text{met1}_\lambda$ -minimaux, alors la valeur de  $\lambda$  est calculable de la façon suivante :

$$\begin{aligned} \text{met1}_\lambda(p_{\text{met1}}) = \text{met1}_\lambda(p_{\text{met2}}) &\iff \text{met1}(p_{\text{met1}}) + \lambda(\text{met2}(p_{\text{met1}}) - \Delta_{\text{met2}}) = \text{met1}(p_{\text{met2}}) + \\ &\quad \lambda(\text{met2}(p_{\text{met2}}) - \Delta_{\text{met2}}) \\ &\iff \lambda = \frac{\text{met1}(p_{\text{met1}}) - \text{met1}(p_{\text{met2}})}{\text{met2}(p_{\text{met2}}) - \text{met2}(p_{\text{met1}})} \end{aligned}$$

**Description de l'algorithme :** L'algorithme de la méthode RLMA (Relaxation Lagrangienne basée sur deux Métriques Additives) est similaire en termes de nombre d'étapes et la manière de fonctionnement avec celui de l'algorithme nommé **méthode de Newton** [70] avec une recherche binaire d'une solution dans un intervalle  $[t_1, t_2]$ . Cet algorithme consiste en trois grandes phases :

- Dans cette phase, la valeur de  $\lambda$  est positionnée à zéro ( $\lambda = 0$ ) et l'algorithme de Dijkstra est appliqué afin de trouver le plus court chemin  $p_{\text{met1}}$  entre le nœud source  $s$  et le nœud destinataire  $t$  en fonction de  $\text{met1}_\lambda$ . C'est-à-dire, le premier plus court chemin trouvé est un plus court chemin en fonction de la métrique  $\text{met1}$ . Si le chemin trouvé vérifie la contrainte imposée par la métrique  $\text{met2}$ , i.e.,  $\text{met2}(p_{\text{met1}}) \leq \Delta_{\text{met2}}$ , alors  $p_{\text{met1}}$  est le meilleur chemin.

- Si le plus court chemin  $p_{\text{met1}} > \Delta_{\text{met2}}$ , le chemin  $p_{\text{met1}}$  est considéré comme le meilleur chemin qui ne satisfait pas la contrainte de met2. L'algorithme de la méthode RLMA doit assurer l'existence d'une solution : il calcule le plus court chemin (nommé  $p_{\text{met2}}$ ) en fonction de la métrique met2. Si  $\text{met2}(p_{\text{met2}}) > \Delta_{\text{met2}}$ , il est clair qu'il n'existe aucun chemin possible entre  $s$  et  $t$  respectant la contrainte  $\Delta_{\text{met2}}$  et l'algorithme s'arrête. Sinon ( $\text{met2}(p_{\text{met2}}) \leq \Delta_{\text{met2}}$ ), le chemin  $p_{\text{met2}}$  est considéré comme le meilleur chemin trouvé jusqu'à maintenant.
- Dans cette phase, les valeurs optimales de  $\lambda$  sont obtenues par des mises à jours répétitives des chemins  $p_{\text{met1}}$  et  $p_{\text{met2}}$  en utilisant d'autres chemins. Soit  $p_{\text{met1}}$  et  $p_{\text{met2}}$  les deux chemins courants. La meilleure valeur de  $\lambda$  qui maximise la fonction  $L(\lambda)$  est  $\frac{\text{met1}(p_{\text{met1}}) - \text{met1}(p_{\text{met2}})}{\text{met2}(p_{\text{met2}}) - \text{met2}(p_{\text{met1}})}$ . Cette valeur de  $\lambda$  sert à trouver un chemin  $p_{\text{update}}$  qui est  $\text{met1}_\lambda$ -minimal en appliquant l'algorithme de Dijkstra. Ce chemin est un candidat pour la solution optimale. Vu que  $\text{met1}_\lambda(p_{\text{met1}}) = \text{met1}_\lambda(p_{\text{met2}})$  (l'étape 5) et si  $\text{met1}_\lambda(p_{\text{update}}) = \text{met1}_\lambda(p_{\text{met1}})$  alors  $p_{\text{met1}}$  et  $p_{\text{met2}}$  sont  $\text{met1}_\lambda$ -minimaux, et donc  $\lambda = \lambda^*$  résultant que le chemin  $p_{\text{met2}}$  est le chemin entre  $s$  et  $t$ . Si  $\text{met1}_\lambda(p_{\text{update}}) \neq \text{met1}_\lambda(p_{\text{met1}})$ , le chemin  $p_{\text{update}}$  remplace  $p_{\text{met1}}$  ou  $p_{\text{met2}}$  selon la contrainte imposée par met2.

L'algorithme LRMA appelé pour trouver un chemin entre deux nœuds  $s$  et  $t$  dans un graphe topologique pondéré  $(G, V, E, \Omega)$  est le suivant :

**LRMA**( $G, s, t$ )

**Étape 1 :**  $p_{\text{met1}} := \text{Dijkstra}(s, t, \text{met1})$  ;

**Étape 2 :** Si  $\text{met2}(p_{\text{met1}}) \leq \Delta_{\text{met2}}$  alors  
 $p_{\text{met1}}$  est le meilleur chemin. Arrêter.

**Étape 3 :**  $p_{\text{met2}} := \text{Dijkstra}(s, t, \text{met2})$  ;

**Étape 4 :** Si  $\text{met2}(p_{\text{met2}}) > \Delta_{\text{met2}}$  alors  
il n'y a pas de chemin possible. Arrêter.

**Étape 5 :**  $\lambda := \frac{\text{met1}(p_{\text{met1}}) - \text{met1}(p_{\text{met2}})}{\text{met2}(p_{\text{met2}}) - \text{met2}(p_{\text{met1}})}$  ;

**Étape 6 :**  $p_{\text{update}} := \text{Dijkstra}(s, t, \text{met1}_\lambda)$  ;

**Étape 7 :** Si  $\text{met1}_\lambda(p_{\text{update}}) = \text{met1}_\lambda(p_{\text{met1}})$  alors  
 $p_{\text{met2}}$  est le meilleur chemin. Arrêter.  
sinon,  
Si  $\text{met2}(p_{\text{update}}) \leq \Delta_{\text{met2}}$  alors  
 $p_{\text{met2}} := p_{\text{update}}$  ;  
sinon,  
 $p_{\text{met1}} := p_{\text{update}}$  ;

**Étape 7 :** Aller à l'Étape 5 ;

**Un chemin optimal sans circuits :** Il est très important que le chemin optimal trouvé par la méthode RLMA ne possède pas de circuit. L'existence d'un circuit dans une route augmente la probabilité de perte, le délai de bout-en-bout, etc.

**Théorème 4.5.2** *Les chemins optimaux trouvés par l'algorithme RLMA sont sans circuits.*

PREUVE :

La démonstration du théorème 4.5.1 peut se faire par contradiction. Supposons que les nœuds  $a$  et  $b$  sont impliqués dans un circuit pour une destination  $t$  (figure 4.3).

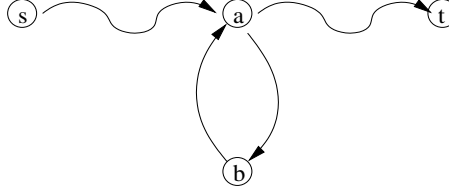


FIG. 4.3 –  $a$  et  $b$  impliqués dans un circuit.

Soit  $s-a-b-t$  le chemin optimal trouvé par l'algorithme RLMA.

Le chemin  $s-a-b-t$  est le plus court chemin en terme de  $\text{met1}_\lambda$ , alors par définition

$$\forall p \neq (s-a-b-t) \in P(s,t), \text{met1}_\lambda(p) > \text{met1}_\lambda(s-a-b-t). \quad (4.29)$$

Puisque  $\text{met1}(s-a-t) \leq \text{met1}(s-a-b-t)$  et  $\text{met2}(s-a-t) \leq \text{met2}(s-a-b-t)$ , alors

$$\text{met1}(s-a-t) + \lambda \times (\text{met2}(s-a-t) - \Delta_{\text{met2}}) \leq \text{met1}(s-a-b-t) + \lambda \times (\text{met2}(s-a-b-t) - \Delta_{\text{met2}}). \quad (4.30)$$

Par conséquent

$$\text{met1}_\lambda(s-a-t) \leq \text{met1}_\lambda(s-a-b-t). \quad (4.31)$$

Si on suppose que  $p = s-a-t$ , les équations (4.29) et (4.31) sont en contradiction.

◆

**Le temps d'exécution de l'algorithme :** Soient  $p_{\text{met1}}^1, p_{\text{met1}}^2, p_{\text{met1}}^3, \dots$  et  $p_{\text{met2}}^1, p_{\text{met2}}^2, p_{\text{met2}}^3, \dots$  les séquences des paquets générés par l'algorithme RLMA. Ces chemins vérifient

$$\begin{cases} \text{met2}(p_{\text{met1}}^1) > \text{met2}(p_{\text{met1}}^2) > \text{met2}(p_{\text{met1}}^3) > \dots > \Delta_{\text{met2}} \\ \text{met2}(p_{\text{met2}}^1) < \text{met2}(p_{\text{met2}}^2) < \text{met2}(p_{\text{met2}}^3) < \dots \leq \Delta_{\text{met2}} \end{cases} \quad (4.32)$$

Puisque le nombre de chemins entre le nœud source et le nœud destinataire est limité, l'algorithme RLMA trouve la valeur optimale  $\lambda$  en un nombre fini d'étapes. Le théorème suivant montre que le temps d'exécution de notre algorithme est polynomial et la démonstration est similaire à celle utilisé dans [70] avec une recherche binaire.

**Théorème 4.5.3** *L'algorithme RLMA se termine après  $O(m \log^3 m)$  itérations, donc le temps d'exécution de l'algorithme est de  $O(m^2 \log^4 m)$ .*

PREUVE :

On prend  $O(m \log n)$  comme le temps d'exécution de l'algorithme de Dijkstra, et on suppose que  $n \ll m$ . Puisque  $n \ll m$ , alors le temps d'exécution de Dijkstra devient  $O(m \log m)$ . Les étapes 1 et 3 prennent un temps  $O(m \log m)$ . Les étapes 2 et 4 prennent un temps  $O(1)$ . Les étapes 5, 6, 7 et 8 se répètent  $O(m \log^3 m)$  fois selon [69] avec une recherche binaire. L'étape 5 s'exécute en  $O(1)$ , l'étape 6 en  $O(m \log m)$ , l'étape 7 en  $O(1)$  et l'étape 8 en  $O(1)$ . Donc l'ensemble des étapes 5, 6, 7 et 8 s'exécutent en  $O(m^2 \log^4 m)$ , vu que  $m \log m \leq m^2 \log^4 m$ .



**L'optimalité des chemins :** Comme nous l'avons déjà précisé, l'optimalité de  $\lambda$  n'implique pas l'optimalité du chemin trouvé à cette valeur. En fait, il n'y a aucune garantie de trouver le chemin optimal, ou une certaine borne supérieure à la borne obtenue. Il y a un cas où l'algorithme RLMA ne trouve pas la solution optimale.

En effet, l'algorithme pourrait trouver la solution optimale, mais il n'est pas sûr de l'avoir trouvé. La figure 4.4 présente un exemple où ce scénario peut se produire en supposant que  $\Delta_{\text{met2}} = 4$ .

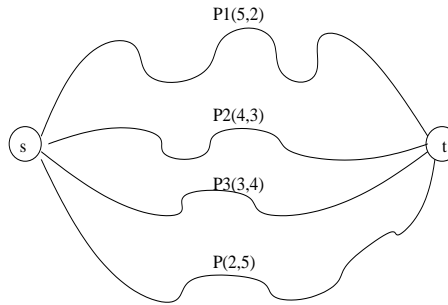
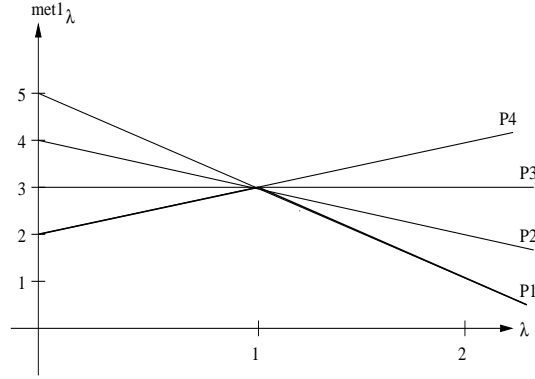


FIG. 4.4 – Un graphe avec 4 chemins entre  $s$  et  $t$ .

Le premier chemin trouvé par l'algorithme RLMA est  $p_4$  qui est  $\text{met1}_0$ -minimal. Puisque  $\text{met2}(p_4) = 5 > \Delta_{\text{met2}} = 4$ , l'algorithme RLMA calcule le plus court chemin en terme de  $\text{met2}$  et il obtient  $p_1$ . Avec  $\text{met2}(p_1) = 2 < \Delta_{\text{met2}} = 4$ , une solution existe. L'étape 5 de l'algorithme trouve  $\lambda = 1$ . Maintenant, l'algorithme calcule le chemin  $P_{\text{Update}}$  en utilisant Dijkstra avec  $\lambda = 1$ . Il obtient 4 chemins (voir figure 4.5) qui ont la même valeur  $\text{met1}_\lambda = 3$  ( $p_1 : 5 + 1 \times (2 - 4) = 3$ ,  $p_2 : 4 + 1 \times (3 - 4) = 3$ ,  $p_3 : 3 + 1 \times (4 - 4) = 3$ ,  $p_4 : 2 + 1 \times (5 - 4) = 3$ ). Dans ce cas là, l'optimalité dépend du chemin choisi par l'algorithme de Dijkstra. La première idée est de choisir le chemin qui minimise  $\text{met1}$  entre ces chemins et qui respecte la contrainte imposée par  $\text{met2}$ . Mais cette solution ne conduit pas forcément à la solution optimale.

## Deux métriques additives et une métrique concave

Il est clair que le problème de trouver un chemin optimal entre deux nœuds en minimisant une des métriques additives ( $\text{met1}$ ) sous les contraintes suivantes : l'autre métrique additive ( $\text{met2}$ ) est inférieure à un seuil et la métrique concave ( $\text{met3}$ ) est supérieure à un seuil est un problème NP-complet vu la présence de deux métriques additives. Dans ce cas là, nous adaptons l'algorithme

FIG. 4.5 – Plusieurs chemins ont la même valeur  $met1_\lambda$ 

RLMA présenté dans la section 4.5.2. Cette adaptation est basée sur la supposition que la métrique concave est la métrique la plus prioritaire à satisfaire. Ce scénario est très réel en prenant la bande passante comme la métrique concave, le délai et le nombre de sauts comme les deux métriques additives. La bande passante a plus de priorité et d'importance que le délai et le nombre de sauts malgré la corrélation forte entre ces métriques.

Soient  $\Delta_{met2}$  et  $\Delta_{met3}$  deux constantes. La représentation formelle du problème à optimiser est la suivante

$$\begin{array}{ll}
 \text{Minimiser} & met1(p) \\
 \text{sous les contraintes} & met2(p) \leq \Delta_{met2} \\
 & met3(p) \geq \Delta_{met3} \\
 & p \in P(s, t),
 \end{array} \tag{4.33}$$

$P(s, t)$  est l'ensemble des chemins entre la source  $s$  et la destination  $t$ . Puisque la métrique  $met3$  est la métrique la plus prioritaire, la première étape pour résoudre le problème (4.33) consiste à réduire l'ensemble de départ ( $P(s, t)$ ) des chemins possibles entre  $s$  et  $t$  en un ensemble de chemins  $P'(s, t)$  vérifiant la contrainte  $met3(p) \geq \Delta_{met3}$  pour tout  $p \in P'$ . Le problème (4.33) devient

$$\begin{array}{ll}
 \text{Minimiser} & met1(p) \\
 \text{sous les contraintes} & met2(p) \leq \Delta_{met2} \\
 & p \in P'(s, t),
 \end{array} \tag{4.34}$$

où  $\forall p \in P'(s, t), met3(p) \geq \Delta_{met3}$ .

Afin de trouver l'ensemble des chemins  $P'(s, t)$ , on utilise le théorème suivant :

**Théorème 4.5.4**  $\forall p = (a_0, \dots, a_n) \in P(s, t)$  où  $a_0 = s$  et  $a_n = t$ ,  $met3(p) \geq \Delta_{met3}$  ssi  $\forall (a_i, a_{i+1}) \in p$  avec  $0 \leq i \leq n-1$ ,  $met3(a_i, a_{i+1}) \geq \Delta_{met3}$ .

PREUVE :

Soit  $p = (a_0, \dots, a_n) \in P(s, t)$  un chemin entre un nœud source  $a_0 = s$  et nœud destinataire  $a_n = t$ . Démontrer que

$$\forall p = (a_0, \dots, a_n) \in P(s, t), \text{met3}(p) \geq \Delta_{\text{met3}} \iff \forall (a_i, a_{i+1}) \in p, \text{met3}(a_i, a_{i+1}) \geq \Delta_{\text{met3}},$$

revient à démontrer que

$$\begin{cases} \forall p = (s, a_0, \dots, a_n, t) \in P(s, t), \text{met3}(p) \geq \Delta_{\text{met3}} \implies \forall (a_i, a_{i+1}) \in p, \text{met3}(a_i, a_{i+1}) \geq \Delta_{\text{met3}} \\ \forall (a_i, a_{i+1}) \in p, \text{met3}(a_i, a_{i+1}) \geq \Delta_{\text{met3}} \implies \text{met3}(p) \geq \Delta_{\text{met3}} \end{cases}$$

On utilise une démonstration par contradiction pour la première implication. On suppose que  $\text{met3}(p) \geq \Delta_{\text{met3}}$  et qu'il existe un arc  $(a_i, a_{i+1})$  tels que  $\text{met3}(a_i, a_{i+1}) < \Delta_{\text{met3}}$ . Puisque chaque arc dans le chemin  $p$  sauf  $(a_i, a_{i+1})$  est associé une valeur de  $\text{met3}$  supérieure à  $\Delta_{\text{met3}}$ , alors on aura  $\text{met3}(p) = \text{met3}(a_i, a_{i+1}) < \Delta_{\text{met3}}$  et donc une contradiction avec la supposition que  $\text{met3}(p) \geq \Delta_{\text{met3}}$ .

La deuxième implication est évidente. Il est clair que  $\forall (a_i, a_{i+1}) \in p, \text{met3}(a_i, a_{i+1}) \geq \Delta_{\text{met3}} \implies \text{met3}(p) \geq \Delta_{\text{met3}}$ .



Le théorème 4.5.4 montre que les arcs qui ont une valeur  $\text{met3}$  inférieure à  $\Delta_{\text{met3}}$  doivent être exclus du nouvel ensemble  $P'(s, t)$ . Cela peut se faire par un simple parcours de vérification.

Une fois l'ensemble  $P'(s, t)$  trouvé, le problème (4.34) devient un problème à deux métriques additives, et l'algorithme RLMA présenté dans la section 4.5.2 peut être appliqué.

Soit  $(G, V, E, \Omega)$  le graphe topologique pondéré, et  $G' = (V', E', \Omega')$  le graphe topologique résultant de l'élimination des liens qui ont une valeur  $\text{met3}$  inférieure à  $\Delta_{\text{met3}}$  et  $\Omega' = \Omega - \{\text{met3}\}$ . L'algorithme final RLMAC (Relaxation Lagrangienne basée 2 Métriques Additive et une Concave) qui résout le problème (4.33) est le suivant

**LRMAC**( $G, s, t$ ) :

**Étape 1 :**  $V' = \emptyset, E' = \emptyset$  ;

**Étape 2 :** Pour chaque arc  $(i, j) \in E$  faire :

si  $\text{met3}(i, j) \geq \Delta_{\text{met3}}$  alors  
 $E' = E' \cup (i, j)$  ;  $V' = V' \cup \{i, j\}$  ;

**Étape 3 :** Si  $(E' = \emptyset)$  alors

il n'y a pas de chemin possible. Arrêter.  
sinon,  
appeler l'algorithme RLMA( $G, s, t$ ).

Les étapes 1 et 2 permettent de construire le sous graphe  $G'$ . Si le graphe  $G'$  est vide alors il n'y a pas de chemin possible entre  $s$  et  $t$ . Autrement, l'algorithme RLMA est appliqué.

Le théorème suivant montre l'absence d'un circuit dans les chemins trouvés par l'algorithme LRMAC.

**Théorème 4.5.5** *Les chemins trouvés par l'algorithme LRMAC sont sans circuits.*

PREUVE :

Les étapes 1 et 2 ne génèrent aucun circuit. En utilisant le théorème 4.5.2, les chemins trouvés par l'étape 3 sont sans circuits. Par conséquent, les chemins trouvés par l'algorithme RLMAC sont sans circuits.



Cet algorithme a le même temps d'exécution que l'algorithme RLMA. En effet, l'étape 1 prend un temps  $O(1)$ , l'étape 2 s'exécute en un temps  $O(m)$  et l'étape 3 en un temps  $O(m^2 \log^4 m)$  (voir le théorème 4.5.3). Alors le temps total d'exécution est de  $O(m^2 \log^4 m)$ .

### Deux métriques additives, une métrique concave et une métrique multiplicative

Nous considérons la probabilité de perte ( $L_p$ ) comme la métrique multiplicative puisqu'elle est la plus utilisée. Donc on aura le problème NP-complet suivant : trouver le plus court chemin entre un nœud source  $s$  et un nœud destinataire  $t$  qui minimise une métrique additive ( $met1$ ), l'autre métrique additive ( $met2$ ) étant inférieure à un seuil ( $\Delta_{met2}$ ), la métrique concave ( $met3$ ) étant supérieure à un seuil ( $\Delta_{met2}$ ) et la probabilité de perte ( $L_p$ ) étant inférieure à un seuil ( $\Delta_{Lp}$ ). Plus formellement

$$\begin{array}{ll}
 \text{Minimiser} & met1(p) \\
 \text{sous les contraintes} & met2(p) \leq \Delta_{met2} \\
 & met3(p) \geq \Delta_{met3} \\
 & Lp(p) \leq \Delta_{Lp} \\
 & p \in P(s, t),
 \end{array} \tag{4.35}$$

$P(s, t)$  est l'ensemble des chemins entre la source  $s$  et la destination  $t$ .

Afin de résoudre le problème (4.35), nous utilisons dans un premier temps la même technique utilisée dans la section 4.5.2. Par la supposition que la métrique  $met3$  est la plus importante et prioritaire, une recherche dans le graphe topologique pour éliminer tous les liens qui n'ont pas assez de  $met3$  est nécessaire. Après cette phase, le problème (4.35) devient

$$\begin{array}{ll}
 \text{Minimiser} & met1(p) \\
 \text{sous les contraintes} & met2(p) \leq \Delta_{met2} \\
 & Lp(p) \leq \Delta_{Lp} \\
 & p \in P'(s, t),
 \end{array} \tag{4.36}$$

où  $\forall p \in P'(s, t)$ ,  $met3(p) \geq \Delta_{met3}$ . On peut utiliser la méthode de la relaxation lagrangienne pour résoudre le problème (4.36) en introduisant deux multiplicateurs de lagrange  $\lambda_1$  et  $\lambda_2$ . La nouvelle fonction objectif à minimiser est  $met1_{\lambda_1 \lambda_2} = met1(p) + \lambda_1(met2(p) - \Delta_{met2}) + \lambda_2(Lp(p) - \Delta_{Lp})$ . Dans ce cas, l'algorithme est un peu compliqué à cause de la non linéarité de la fonction  $met1_{\lambda_1 \lambda_2}$ . Pour éviter cet obstacle, on utilise une métrique unique mixte qui combine  $met2$  et  $Lp$ . Au lieu de la probabilité de perte, on emploie le logarithme de la probabilité de transmission avec succès ( $slog$ ) pour éviter une règle de composition complexe. i.e., pour la probabilité de succès de transmission ( $Stp$ ), on a  $Stp(p) = \prod_{i=1}^k Stp(v_{i-1}, v_i)$  alors  $slog(Stp(p)) = \sum_{i=1}^k slog(Stp(v_{i-1}, v_i))$ . Par ailleurs, nous supposons que les routes ne peuvent pas avoir une probabilité de perte plus grande que 90%. De ce fait, on conclut que  $0 < |slog| < 1$ . Nous considérons que l'unité milliseconde est appropriée

pour le délai [71] et nous nous limitons donc au cas où le délai est un chiffre entier. Par conséquent, nous avons une métrique unique qui présente les métriques  $\text{met2}$  et la probabilité de perte, où la partie entière est  $\text{met2}$  et la partie décimale représente la probabilité de perte. Cette propriété nous permet de construire une heuristique simple et capable de découvrir des chemins praticables dans le réseau.

En fusionnant les deux métriques,  $\text{met2}$  et la probabilité de perte ( $L_p$ ) sont remplacées par une métrique unique mixte ( $\text{um}$ ) dans l'ensemble  $\Omega$  tels que  $\text{um}(i, j) = \text{met2}(i, j) + \text{slog}(i, j)$ , où  $\text{slog}(i, j)$  est le logarithme de la probabilité de transmission avec succès entre les nœuds  $i$  et  $j$ . Les deux constantes  $\Delta_{\text{met2}}$  et  $\Delta_{L_p}$  sont substituées par  $\Delta_{\text{um}}$  tel que  $\Delta_{\text{um}} = \Delta_{\text{met2}} + \text{slog}(1 - \Delta_{L_p})$ . La métrique unique mixte ( $\text{um}$ ) est additive parce que la somme de deux métriques additives est additive. Afin d'assurer que la somme des parties décimales des arcs d'un chemin ne dépasse pas la valeur  $\text{slog}(1 - \Delta_{L_p})$ , des vérifications sont nécessaires.

Le passage à une seule métrique unique mixte permet de simplifier le problème (4.36) au problème suivant :

$$\begin{array}{ll} \text{Minimiser} & \text{met1}(p) \\ \text{sous les contraintes} & \text{um}(p) \leq \Delta_{\text{um}} \end{array} \quad (4.37)$$

où  $\forall p \in P'(s, t)$ ,  $\text{met3}(p) \geq \Delta_{\text{met3}}$ . Le problème (4.37) devient un problème à deux métriques additives, et l'algorithme RLMA présenté dans la section 4.5.2 peut être employé mais en ajoutant le test suivant : la somme des parties décimales des arcs de chaque chemin sélectionné selon les valeurs des  $\lambda$  trouvées n'est pas supérieure à  $\text{slog}(1 - \Delta_{L_p})$ . Ces vérifications sont rajoutées dans l'algorithme de Dijkstra. En effet, le plus court chemin doit posséder une somme des parties décimales de ses arcs inférieure à  $\text{slog}(1 - \Delta_{L_p})$ .

Soit  $(G, V, E, \Omega)$  le graphe topologique pondéré et  $G' = (V', E', \Omega')$  le graphe topologique résultant de l'élimination des liens qui ont une valeur  $\text{met3}$  inférieure à  $\Delta_{\text{met3}}$  avec  $\Omega' = \Omega - \{\text{met3}\} \cup \{\text{um}\}$ . L'algorithme final RLMACM (Relaxation Lagrangienne basée 2 Métriques Additive, une Concave et une multiplicative) qui résout le problème (4.35) est le suivant

**LRMACM**( $G, s, t$ ) :

**Étape 1 :**  $V' = \emptyset, E' = \emptyset$  ;

**Étape 2 :** Pour chaque arc  $(i, j) \in E$  faire :

si  $\text{met3}(i, j) \geq \Delta_{\text{met3}}$  alors  
 $E' = E' \cup (i, j) ; V' = V' \cup \{i, j\}$  ;

**Étape 3 :** Si  $(E' = \emptyset)$  alors

il n'y a pas de chemin possible. Arrêter.

sinon,

appliquer l'algorithme RLMA( $G, s, t$ ).

Cet algorithme est exactement le même que RLMA avec des vérifications supplémentaires au niveau de la relaxation lagrangienne. Donc, le temps total d'exécution est de  $O(m^2 \log^4 m)$  et les chemins trouvés par cet algorithme sont sans circuits.



## 4.6 Résultats de simulation

Dans cette section, nous montrons les performances de la méthode RLMA qui est la base de la résolution des problèmes à deux, trois et quatre métriques. Nous générons 100 fois trois graphes aléatoires de 100 nœuds chacun (figure 4.6). Un générateur de graphes aléatoires est utilisé pour créer les liens entre les nœuds d'un même graphe et entre les trois graphes de telle façon que chaque nœud aura au minimum 4 voisins [53].

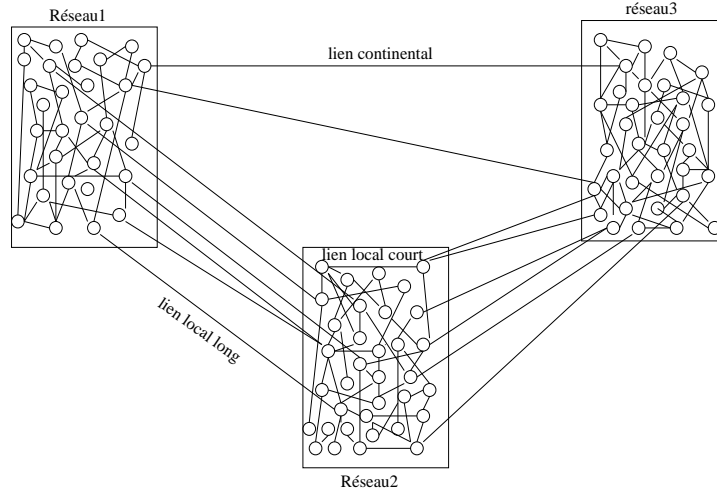


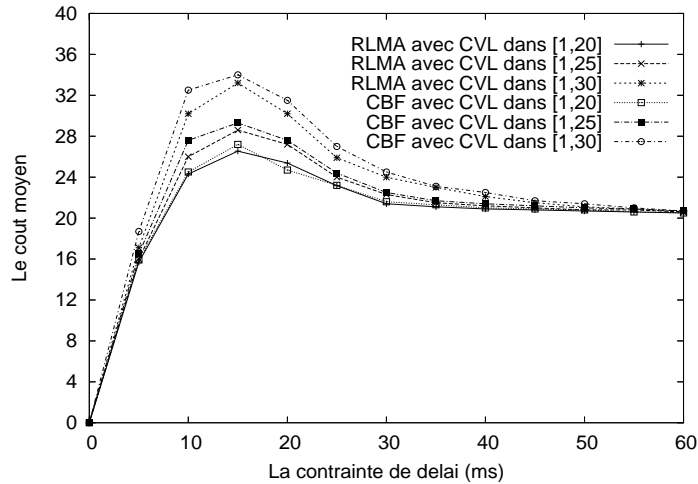
FIG. 4.6 – Le modèle de topologie.

Afin d'étudier les performances de la méthode RLMA, nous considérons le problème de minimiser le coût entre deux nœuds en conservant un délai de propagation inférieur à un seuil. Le coût et le délai de propagation sont deux métriques additives qui remplacent les métriques  $met1$  et  $met2$  respectivement. La valeur de délai de chaque lien inter-graphe est sélectionnée uniformément entre 20 et 30 ms. Au sein d'un même graphe, 20% des liens sont considérés comme des liens locaux à grande distance et par conséquent leurs délais sont sélectionnés entre 8 et 20 ms. Des délais entre 1 et 8 ms sont attribués au reste des liens locaux. Ce schéma de distribution à trois classes de délai modélise une situation réelle d'un réseau entre 3 continents [53].

### 4.6.1 Le coût moyen

La figure 4.7 illustre le coût moyen des chemins trouvés par l'algorithme RLMA comparé avec celui des chemins trouvés par l'algorithme de Bellman-Ford sous contraintes (CBF) [46]. Rappelons que CBF trouve des solutions optimales entre un nœud source et un ensemble des nœuds destinataires avec un temps d'exécution exponentiel.

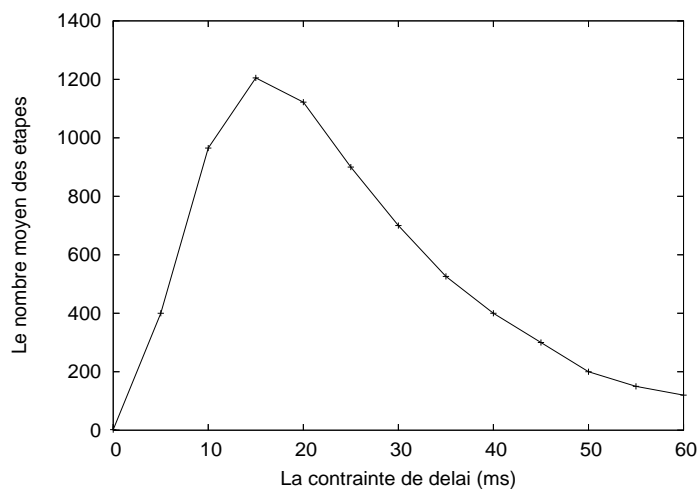
Dans cette figure, nous considérons 3 scénarios où chaque scénario attribue à ces liens des valeurs de coût (CVL : *Cost Value en Link*) entre 1 et 20, 1 et 25 ou 1 et 30. Les chemins trouvés par notre algorithme présentent pratiquement les mêmes performances que ceux trouvés par l'algorithme CBF. Ceci montre l'efficacité de la méthode RLMA. Pour  $\Delta_{del} < 15$  ms, il n'y a pas de garantie de trouver un plus court chemin entre deux paires de nœuds quelconques avec un délai inférieure à  $\Delta_{del}$  et par conséquent le coût moyen des chemins trouvés est petit. Avec l'augmentation de la borne  $\Delta_{del}$ , l'algorithme RLMA trouve plus de chemins avec plus de sauts ce qui augmente

FIG. 4.7 – *Le coût moyen.*

la valeur moyenne de coût. Avec  $\Delta_{\text{del}} = 15$  ms, chaque nœud dans le réseau trouve au moins un chemin vers tous les autres nœuds en utilisant notre algorithme. Lorsque  $\Delta_{\text{del}} > 15$  ms, notre algorithme est capable de trouver des chemins minimaux en coût avec un des larges délais et donc la valeur moyenne de coût diminue avec l'augmentation de la borne de délai  $\Delta_{\text{del}}$ .

#### 4.6.2 Le nombre moyen d'étapes

Le nombre moyen d'étapes de l'algorithme RLMA est un élément important pour déterminer le temps moyen d'exécution. Le nombre moyen d'étapes est déterminé par le nombre de fois que l'algorithme de Dijkstra est exécuté. La figure 4.8 est obtenue en modifiant le seuil (la borne supérieure) du délai.

FIG. 4.8 – *Le nombre moyen des étapes.*

Pour  $\Delta_{del} < 15$  ms, l'algorithme RLMA trouve de plus en plus des chemins et par conséquent, le nombre d'exécution de l'algorithme de Dijkstra augmente. Avec  $\Delta_{del} = 15$  ms, l'algorithme de Dijkstra est appelé 1200 fois ce qui représente le nombre maximal d'itérations. En dépassant les 15 ms, l'algorithme RLMA peut trouver la solution optimale après une seule itération d'algorithme de Dijkstra.

### 4.6.3 Comparaison des performances

Dans cette partie, les performances de notre algorithme RLMA sont comparées avec la deuxième approximation de l'algorithme de Jaffe [59], l'heuristique de Chen [55] et l'algorithme  $\varepsilon$ -optimal [47]. Dans [59], Jaffe a prouvé que la deuxième approximation présente de meilleures performances par rapport à la première et donc cette approximation sera utilisée comme élément de comparaison. Les solutions exactes sont aussi calculées par un algorithme exhaustif (temps exponentiel) en considérant tous les chemins possibles dans le graphe. Ces solutions permettent de déterminer l'optimalité des chemins trouvés par notre algorithme. Dans notre modèle de simulation, le réseau est considéré comme un graphe orienté pondéré. Les poids d'un arc, les seuils  $\Delta_{met2}$  pour chaque connexion, la source et la destination d'une requête de connexion sont générées aléatoirement. Nous utilisons le taux de succès (SR : *Success Ratio*) pour illustrer la probabilité qu'un algorithme trouve un chemin optimal et faisable. SR est donné par la formule suivante :

$$SR = \frac{\text{Nombre total de requêtes de connexion réussies}}{\text{Nombre total de requêtes de connexion}}.$$

Une connexion est considérée comme réussie si l'algorithme trouve un chemin faisable et optimal pour cette connexion. Nous considérons le graphe topologique de la figure 4.9. Le même graphe est utilisé dans [55].

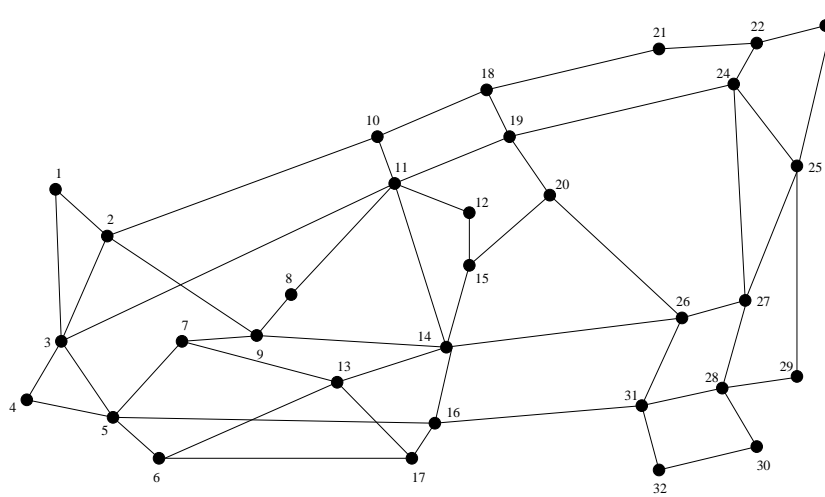


FIG. 4.9 – Graphe topologique.

Les valeurs de  $\Delta_{met1}$  et  $\Delta_{met2}$  sont tirées uniformément dans  $[0, 50]$  et  $[0, 200]$  respectivement. La table 4.1 est obtenue en variant les intervalles de tirage du seuil  $\Delta_{met2}$ . Cette table montre le SR de plusieurs algorithmes calculés sur 20 exécutions et chaque exécution comporte 2000 requêtes de connexion générées aléatoirement.

$\Delta_{\text{met}2}$	Exhaustif	RLMA	Jaffe	Chen, $x = 2$	Chen, $x = 10$	1-optimal	10-optimal
[200,260]	0.2594	0.2592	0.2505	0.1935	0.2554	0.2524	0.2099
[300,360]	0.5220	0.5201	0.4906	0.3004	0.5003	0.5057	0.3479
[400,460]	0.7595	0.7588	0.7088	0.3308	0.7216	0.7430	0.4703
[500,560]	0.9219	0.9200	0.8674	0.3308	0.8787	0.9079	0.5639
[600,660]	0.9868	0.9799	0.9524	0.3308	0.9609	0.9805	0.6281

TAB. 4.1 – Les SR de plusieurs algorithmes.

L'augmentation de la borne  $\Delta_{\text{del}}$  permet aux algorithmes de trouver plus de chemins faisables ce qui explique l'augmentation des valeurs de SR. L'algorithme RLMA présente pratiquement les mêmes performance que l'algorithme exhaustif et de meilleures performances que l'algorithme de Jaffe. Une comparaison entre l'algorithme RLMA, Chen et  $\varepsilon$ -optimal n'est possible qu'avec des valeurs précises de  $x$  et  $\varepsilon$ . Théoriquement, lorsque  $x$  tend vers l'infini et  $\varepsilon$  vers 0, les performances des algorithmes correspondants approchent de celles de l'algorithme exhaustif. Cependant, de larges valeurs de  $x$  et petites valeurs de  $\varepsilon$  augmentent significativement la complexité et leurs algorithmes correspondants deviennent impraticables. Si les algorithmes de Chen et  $\varepsilon$ -optimale ont la même complexité moyenne de temps de calcul que celui de RLMA, une étude comparative sera possible. Cette égalité en termes complexité est obtenue lorsque  $x = 2$  et  $\varepsilon = 10$ . Avec  $x = 2$ , l'algorithme de Chen trouve moins de chemins que notre algorithme avec la même complexité de calcul. Lorsque  $x = 10$ , la complexité du calcul de l'algorithme de Chen augmente avec une amélioration sensible du nombre de chemins trouvés mais les SR restent toujours légèrement inférieurs à ceux de notre algorithme. L'algorithme 10-optimal ( $\varepsilon = 10$ ) a la même complexité de calcul que celle de RLMA mais il est 50% moins performant en termes de SR. Les performances de l'algorithme  $\varepsilon$ -optimal devient similaire à notre algorithme lorsque  $\varepsilon = 1$ .

## 4.7 Conclusion

Dans le routage classique, le calcul de la route est basé sur une seule métrique en appelant un des algorithmes de plus court chemin. Lorsque le nombre des métriques augmente, le problème du routage devient NP-complet et les chemins trouvés ne sont pas toujours optimaux. Dans un premier temps, nous avons proposé un algorithme linéaire qui choisit un plus court chemin parmi plusieurs chemins de même largeur. Cet algorithme favorise la métrique concave sur les autres métriques et utilise l'algorithme de Dijkstra comme principe de base. Pour des problèmes de routage plus complexe, nous avons proposé un algorithme pouvant traiter deux, trois et quatre métriques en un temps quasi-linéaire ( $O(m^2 \log^4 m)$ ) en se basant sur la technique de la relaxation lagrangienne. L'algorithme RLMA développé pour deux métriques additives est considéré comme un noyau pour les autres algorithmes RLMAC ou RLMACM. RLMAC et RLMACM rajoutent des étapes par rapport au RLMA pour réduire l'ensemble de départ des chemins possibles entre la source et la destination et fusionner les métriques dans le but d'avoir deux métriques additives. L'algorithme RLMA présente les mêmes performances que celles des algorithmes exhaustifs qui trouvent des solutions exactes en un temps qui peut être exponentiel.



## Chapitre 5

# Routage ad hoc et qualité de service

Le domaine des réseaux ad hoc est très prometteur puisqu'il permet la création spontanée d'un réseau sans avoir besoin d'aucune infrastructure. Un réseau ad hoc est simplement constitué de terminaux utilisateurs sachant communiquer ensemble directement. Une façon intuitive et simple de concevoir les réseaux ad hoc est de considérer qu'ils correspondent à la généralisation ultime des réseaux sans fil car ils limitent au maximum le rôle de l'infrastructure fixe. Cette généralisation est obtenue par l'amélioration des capacités de connectivité des réseaux locaux sans fil. La portée limitée des terminaux exige la présence d'un protocole de routage pour établir une communication entre deux entités lointaines. Plusieurs protocoles de routage ont été proposés dans le groupe MANET. Ils permettent de trouver des plus courts chemins en termes nombre de sauts.

Comme la Qualité de Service (QoS) est un point important dans tous les réseaux (informatique et telecom), il semble intéressant d'étudier les moyens permettant d'introduire cette notion dans les réseaux ad hoc mobiles (MANETs) où les terminaux sont en mouvement les uns par rapport aux autres. En effet, les applications multimédias que l'on connaît aujourd'hui nécessitent des garanties de QoS plus ou moins importantes qu'il serait bon de pouvoir proposer dans ce type de réseaux encore expérimentaux.

La QoS peut être apportée à différents niveaux : couche application, couche transport, couche réseau, couche MAC, etc. Dans cette thèse, nous nous concentrerons sur des solutions de signalisation et de routage permettant d'assurer une certaine qualité de service dans les réseaux ad hoc mobiles au niveau de la couche réseau.

### 5.1 Les réseaux Ad hoc

Un réseau ad hoc est un ensemble de nœuds autonomes capable de se déplacer et communiquer librement sans aucune infrastructure existante ni contrainte imposée. Ceci résulte en une topologie changeante dans le temps d'une manière imprévisible. D'autre part, la portée des transmissions radio étant limitée, le relayage est rendu obligatoire, et il faut donc que les nœuds formant ce réseau ad hoc coopèrent pour retransmettre les messages d'une source vers une destination. Les chemins utilisés et les nœuds traversés sont déterminés par un protocole de routage dédié.

Les protocoles de routage doivent prendre en compte les caractéristiques de ces réseaux mobiles, multi-sauts et sans fil. Les difficultés de conception sont principalement liées (i) aux caractéristiques spécifiques des liens sans fil et (ii) aux changements fréquents de topologie dus à la mobilité.

### 5.1.1 Routage niveau MAC vs niveau réseau

Le routage dans un réseau mobile ad hoc peut se situer à différentes couches. Le HiperLAN1 préconise un routage au niveau MAC, alors que MANET (*Mobile Ad hoc NETwork*) [3] prévoit un routage au niveau IP. D'autres systèmes comme ANANAS [72] sont en faveur d'un routage au niveau intermédiaire.

#### HiperLAN1

HiperLAN1 peut être considéré comme l'ancêtre du protocole OLSR (*Optimized Link State Routing protocol*) [73]. Il offre la possibilité aux nœuds HiperLAN1 de communiquer entre eux, même s'ils ne sont pas à portée radio et cela grâce au relaiage. Le relaiage est effectué dans la couche MAC. Afin d'optimiser l'utilisation de la bande passante, la technique des relais multipoints est proposé pour permettre la diffusion des trames en utilisant un sous-ensemble de voisins plutôt que tous les voisins. Cette technique a été reprise par OLSR et appliquée au niveau IP. Elle sera détaillée plus loin. Les tables de relaiage sont construites à partir des informations de voisinage et de topologie diffusées périodiquement dans le réseau. Les trames de contrôle ont une priorité plus élevée que les données, et les trames de topologie se propagent dans tout le réseau en utilisant les relais multipoint. Un réseau HiperLAN1 est vu par les couches hautes (couches au niveau supérieur ou égale à 3) comme un réseau local à diffusion.

A défaut de cartes dédiées HiperLAN, l'implémentation de routage ad hoc au niveau MAC nécessite une modification des pilotes des cartes sans fil qui n'est pas toujours possible (plusieurs types de cartes existent sur le marché et le code source des drivers n'est pas toujours fourni).

#### MANET

MANET [3] est un groupe de travail créé au sein de l'IETF (*Internet Engineering Task Force*) [4] dont le but est de spécifier et de standardiser des protocoles de routage pour les réseaux ad hoc au niveau IP. Ces protocoles doivent pouvoir supporter des couches physique et MAC hétérogènes et offre aux couches supérieures la possibilité de fonctionner uniquement sur IP. La figure 5.1 illustre un exemple de réseau MANET utilisant deux technologies sans fil différentes (*A* et *B*) ; l'union de ces deux topologies physiques forme une seule topologie logique qui sera utilisée par le routage IP. Ceci diffère de la solution HiperLAN1 qui spécifie et fige les couches basses du système.

Le document [74] définit un réseau MANET comme *un réseau de plates-formes mobiles autonomes qui peuvent avoir plusieurs hôtes et interfaces de communication. Ces plates-formes sont libres de bouger sans contraintes et doivent pouvoir fonctionner en réseau autonome et supporter des liaisons vers des réseaux fixes via des passerelles. Les réseaux MANET auront des topologies dynamiques à sauts multiples et une taille variant de quelques dizaines à des centaines de nœuds mobiles.*

Le premier objectif du groupe est de retenir un ou plusieurs protocoles de routage unicast et définir l'interaction avec les couches supérieures et inférieures. Ensuite, d'étudier les problèmes de la qualité de service et le multicast dans un environnement mobile MANET. Une partie importante de cette thèse traite la qualité de service dans les réseaux ad hoc.

### 5.1.2 Les protocoles de routage pour les réseaux Ad Hoc

L'objectif du routage MANET est de trouver la route la plus courte en terme de nombre de sauts vers une destination. Les protocoles de routage des réseaux ad hoc s'appuient sur trois mo-

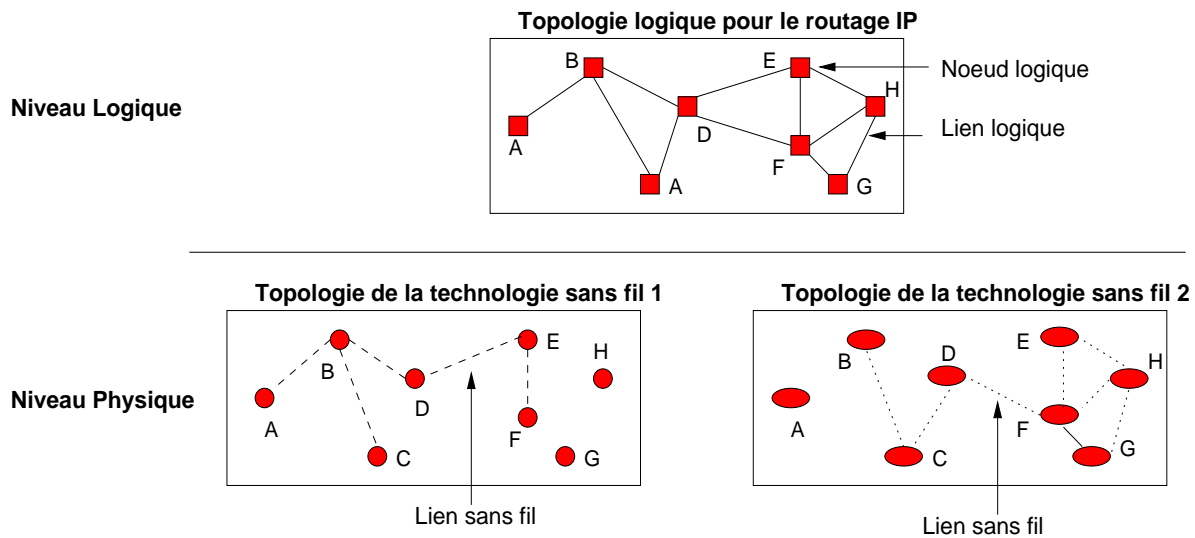


FIG. 5.1 – Exemple de réseau MANET.

dèles de fonctionnement : les protocoles proactifs, les protocoles réactifs et les protocoles hybrides. On peut les différencier par la méthode utilisée pour découvrir le chemin entre le nœud source et le nœud destination.

Les protocoles proactifs établissent et mettent à jour les routes pour tous les nœuds du réseau en se basant sur l'échange périodique d'information de routage. Ces protocoles ont l'avantage de la disponibilité immédiate des routes vers tous les nœuds du réseau. Ainsi, le délai d'acheminement des paquets est très court. Cependant, un trafic de contrôle important est nécessaire pour mettre à jour les routes et converger vers un état cohérent dans un réseau avec une topologie très dynamique. Comme dans les réseaux filaires, deux principales méthodes sont utilisées : le routage par vecteur de distance et le routage par état de lien. Par exemple, Dynamic Destination-Sequenced Distance-Vector (DSDV [75]) est une adaptation du protocole classique Routing Information Protocol (RIP [76]) pour les réseaux ad hoc. Optimized Link State Routing (OLSR [73]) quant à lui est une optimisation de l'algorithme d'état de lien Open Shortest Path First (OSPF [77]) pour les réseaux ad hoc. Le routage par état de lien consiste à diffuser périodiquement l'état des liens des voisins à tous les nœuds de réseau, par contre le routage par vecteur de distance permet à chaque nœud de diffuser à ses voisins sa vision des distances qui lui séparent de tous les nœuds de réseau. Les algorithmes de routages basés sur ces deux méthodes utilisent la même technique qui est la technique des plus courts chemins. L'inconvénient des protocoles proactifs réside dans le coût du maintien des informations de topologie et de routage même en absence de trafic de données ce qui implique une consommation continue de bande passante.

Au contraire des protocoles proactifs, les protocoles réactifs établissent les routes uniquement à la demande. Une procédure de découverte de route est déclenchée lorsqu'un nœud souhaite envoyer des paquets vers un destinataire dont la route est inconnue. Une demande de route vers ce destinataire est propagée dans le réseau. Un nœud ayant une route valide vers ce destinataire, répond au nœud à l'origine de la demande. La route est obtenue par le chemin inverse de la réponse. Citons par exemple les protocoles, Ad hoc On demand Distance Vector (AODV [78]), et Dynamic Source Routing Protocol (DSR [79]). La propagation d'ouverture de la route est basée sur le mécanisme d'inondation. Cette inondation surcharge tout le réseau puisque tous les nœuds doivent répéter la requête. De plus, un problème supplémentaire se pose lors de changement de



topologie : ce type de protocole est obligé de réagir rapidement pour trouver une autre alternative à la route endommagée. En conséquence, un délai d'attente et une bufferisation des paquets de donnée sont des contraintes à gérer. Il est aussi impossible de connaître au préalable la qualité du chemin (en termes de bande passante, délai, etc). Une telle connaissance est importante dans les applications multimédias.

Les protocoles hybrides combinent les deux idées des protocoles proactifs et réactifs. Ils utilisent un protocole proactif, pour apprendre le proche voisinage (par exemple voisinage à deux sauts ou à trois sauts) ; ainsi ils disposent des routes immédiatement dans le voisinage. Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Avec ce découpage, le réseau est partagé en plusieurs zones, et la recherche de route en mode réactif peut être améliorée. À la réception d'une requête de recherche réactive, un nœud peut indiquer immédiatement si la destination est dans le voisinage ou non, et par conséquent savoir s'il faut aiguiller la requête vers les autres zones sans déranger le reste de sa zone. Zone Routing Protocol (ZRP [80]) et Cluster Based Routing Protocol (CBRP [81]) sont des exemples de tel protocole. Ce type de protocole d'adapte bien aux grands réseaux, cependant, il cumule aussi les inconvénients des protocoles réactifs et proactifs : message de contrôle périodiques plus le coût d'ouverture d'une nouvelle route.

À l'heure actuel, seulement trois protocoles : OLSR, AODV, TBRPF (*Topology Dissemination Based on Reverse-Path Forwarding*) [82] sont passé à l'état RFC (*Request For Comments*) de catégorie *Expérimental*. DSR est encore un draft mais il sera bientôt un RFC Expérimental. Les travaux actuels de MANET visent à converger vers un seul protocole prenant les points forts de chaque protocole.

## 5.2 Optimized Link State Routing protocol (OLSR)

Le protocole OLSR [73, 83] est un protocole proactif. Il applique dans un contexte ad hoc les règles de routage, cette fois centrées sur l'état du lien. Dans un protocole de routage par état de lien, tous les liens avec les nœuds voisins sont déclarés et inondés dans le réseau. Cette technique permet à chaque nœud de connaître parfaitement une vision globale sur la topologie du réseau. En utilisant cette carte topologique, un nœud source peut choisir le chemin le plus court vers une destination en appelant l'algorithme Dijkstra, couramment utilisé dans les techniques de découverte de routes d'une source vers une destination. Le protocole OLSR, est une optimisation de l'algorithme d'état de lien pure pour les réseaux Ad hoc : (i) il réduit la taille des messages de contrôle, au lieu de tous les liens, il déclare un sous ensemble de liens avec ses voisins qui sont les relais multipoints (MPR) ; (ii) il minimise le coût d'inondation du trafic de contrôle par l'utilisation seulement des nœuds relais multipoint pour diffuser ses messages. Seuls les MPRs retransmettent les messages diffusés. La technique des relais multipoints réduit significativement le nombre des retransmissions redondantes lors de diffusion.

Les nœuds OLSR échangent périodiquement des messages de contrôle et maintiennent des routes pour atteindre tout nœud OLSR du réseau. le protocole OLSR réalise principalement deux fonctionnalités : Détection de voisinage et Gestion de la topologie.

### 5.2.1 Détection de voisinage

Chaque nœud doit détecter toutes les interfaces de ses voisins ayant un lien direct et symétrique avec l'une de ses interfaces. En raison de l'incertitude de la propagation radio, le lien entre

deux nœuds voisins peut être unidirectionnel. Les liens doivent donc être vérifiés dans les deux sens avant de les considérer comme valides. Alors, chaque nœud doit diffuser périodiquement un message *Hello* dans son voisinage direct (TTL=1). Ce message contient les informations relatives aux interfaces entendues par ce nœud : la liste des adresses des interfaces des nœuds voisins avec leur état de lien. Un lien entre les interfaces d'un nœud et son voisin peut avoir l'un des quatre états suivant : « symétrique », « asymétrique », « MPR » ou « perdu ». Symétrique signifie que le lien est validé comme bidirectionnel est qu'il est possible de transmettre des données dans les deux sens. Asymétrique signifie que le nœud entend cette interface de voisin mais le lien n'est pas encore validé dans l'autre sens. MPR indique que ce nœud a sélectionné ce voisin comme relais multipoint et cela implique que le lien est symétrique. Perdu indique le lien avec cette interface de voisin n'est plus valide.

Les informations de voisinage sont maintenues dans une base d'information concernant les voisins directs et leurs interfaces multiples, les voisins à deux sauts, les relais multipoints et les sélecteurs de relais multipoint. La figure 5.2 montre un exemple de détection de voisinage. Chacun des nœuds possède une seule interface sans fil. Il est nécessaire qu'un nœud possédant plusieurs interfaces, choisisse l'adresse d'une de ses interfaces comme son adresse principale.

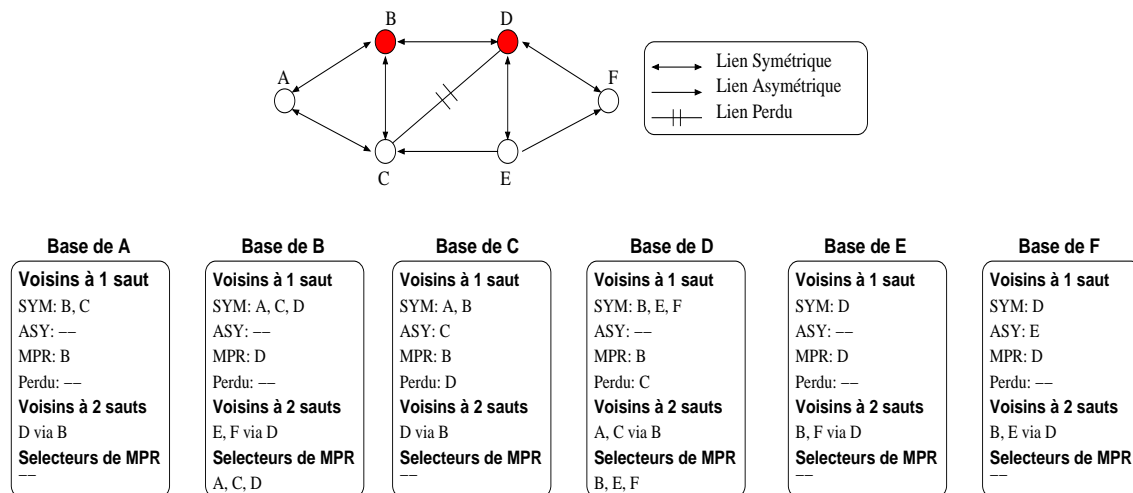


FIG. 5.2 – Exemple d'information de voisinage maintenue par OLSR.

Les messages *Hellos* sont diffusés sur toutes les interfaces du nœud émetteur et ils sont reçus par tous les voisins qui se trouvent à un saut de ce nœud. Ces messages sont envoyés avec une fréquence déterminée par l'intervalle *Hello\_Interval* (la valeur par défaut est de 2 seconde). Les voisins qui reçoivent ces messages, les traitent et ne les relaient pas. Les messages *Hellos* permettent également de découvrir les voisins à deux sauts (c'est-à-dire, les voisins des voisins). Les informations de voisinage fournies par ces messages sont valide pendant une durée de vie *Neighbor\_Hold\_time* qui est égale à  $3 \times \text{Hello\_Interval}$ . Une approche simple permettant de détecter la non-validité d'un lien OLSR est la perte de 3 *Hellos* successifs, mais d'autres techniques peuvent être utilisées dans OLSR.

### 5.2.2 Le concept des relais multipoint

Le concept de relais multipoint vise à diffuser efficacement les messages destinés à tous les nœuds du réseau ad hoc (par exemple, les messages *TC*). La transmission radio étant par défaut

une inondation à tous les voisins directs. Les nœuds à deux sauts d'une source peuvent être joints par une retransmission d'un ou plusieurs voisins directs. L'idée de base est de désigner un nombre suffisant de voisins appelés relais multipoint permettant de réduire le nombre de retransmissions redondantes dans la même région du réseau.

En se basant sur l'information de voisinage, chaque nœud  $s$  sélectionne indépendamment un sous-ensemble minimal de nœuds parmi ses voisins directs pour retransmettre ses paquets. Ces nœuds possèdent des liens symétriques avec  $s$  et leur ensemble est noté  $MPR(s)$ . Les nœuds voisins de  $s$  qui ne sont pas des relais multipoints de ce nœud, reçoivent et traitent les messages diffusés par  $m$ , mais ne les retransmettent pas. Chaque nœud MPR maintient l'ensemble de ses sélecteurs de relais multipoint, noté Multipoint relay Selector (MS) et ne retransmet que les paquets reçu pour la première fois de ses sélecteurs de relais multipoint (voir figure 5.3). Cette technique réduit d'une façon significative le nombre de retransmissions lors du processus de diffusion [84]. Les relais multipoint sont calculés suite à la détection d'un changement de voisinage direct ou à deux sauts. La figure 2.15.a montre un exemple où un paquet de  $s$  est diffusé au voisinage à trois sauts par 24 retransmissions. Dans la figure 2.15.b seuls les relais multipoint retransmettent le paquet (12 retransmissions).

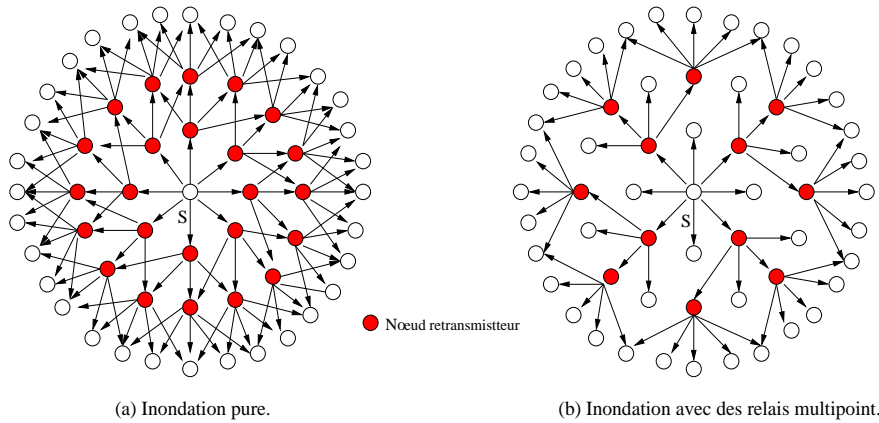


FIG. 5.3 – Optimisation de l'inondation par des relais multipoint.

L'optimisation offerte par l'utilisation des relais multipoint est plus efficace dans des topologies de réseaux ad hoc denses et larges. Par ailleurs, cette optimisation s'avère bénéfique pour la recherche de route par inondation utilisée dans les protocoles réactifs [85]. Le gain sera important dans les deux configurations suivante : (i) pour les modèles de trafic aléatoire et sporadique où un large sous-ensemble de nœuds est en communication, (ii) lorsque les couples [source, destination] varient dans le temps [86]. Plus le nombre de relais multipoint est petit, plus le routage est optimal. Une heuristique pour la sélection d'un ensemble minimal de relais multipoint a été proposée dans [73]. Cette heuristique est bien détaillée dans le chapitre 3.

### 5.2.3 Gestion de la topologie

Chaque nœud sélectionné comme MPR dans le réseau diffuse périodiquement à tous les nœuds du réseau des messages de topologie, appelés Topology Control ( $TC$ ) avec une fréquence déterminée par l'intervalle  $TC\_Interval$ . Un message  $TC$  contient les nœuds ayant sélectionné le nœud  $s$  origine du  $TC$  comme relais multipoint et un numéro de séquence associé à l'ensemble des sélecteurs de relais multipoint qui sera incrémenté à chaque changement de cet ensemble. Uniquement

les nœuds MPRs ont le droit de rediffuser les *TCs*. La réception de ces messages *TCs*, permet à chaque nœud dans le réseau de maintenir une base contenant les informations topologiques du réseau qui donne une vision globale (mais non pas entière) de la topologie du réseau. Cette base est constitué d'un ensemble de tuples  $[T\_dest, T\_last, T\_seq, T\_time]$ .  $T\_dest$  est l'adresse principale de la destination, qui est accessible en un saut à partir du nœud avec l'adresse principale  $T\_last$ . En d'autre terme,  $T\_last$  est un relais multipoint de  $T\_dest$ .  $T\_seq$  est un numéro de séquence et  $T\_time$  spécifie le temps au bout duquel le tuple est expiré. La figure 5.4 montre la base topologique de l'ensemble des nœuds du réseau ad hoc de la figure 5.2.

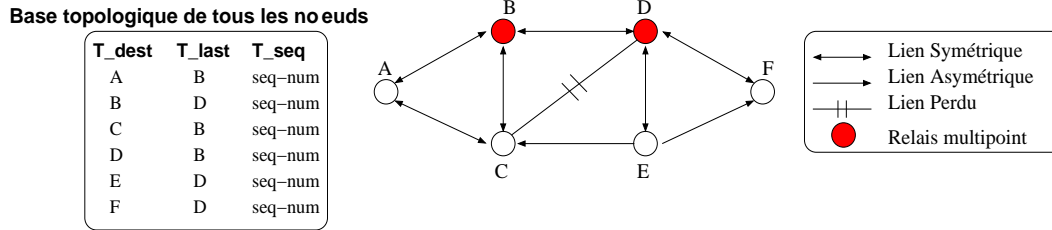


FIG. 5.4 – Exemple d'information de topologie maintenue par OLSR.

#### 5.2.4 Calcul des routes

Le routage se fait saut par saut. Chaque nœud dans le réseau calcule sa table de routage pour atteindre tout autre nœud présent dans le réseau. Ce calcul est basé sur les informations de voisinage et de topologie rafraîchies périodiquement. Un nœud  $s$  utilise l'algorithme de Dijkstra [87], un algorithme du plus court chemin, sur le graphe de topologie partiel construit à partir des paires suivantes :

- Les liens  $(m, v)$  existants dans la table de voisinage, tels que  $v$  est un voisin symétrique de  $m$  ;
- Les liens  $(last, dest)$  existants dans la table de topologie.

Les routes trouvées sont optimales en nombre de sauts, sans boucles et les nœuds intermédiaires sont des relais multipoint. La table de routage est mise à jour chaque fois qu'il y a un changement dans la base de voisinage ou de la topologie. C'est-à-dire, quand on détecte l'apparition ou la perte d'un voisin ou lors de l'ajout ou la suppression d'un tuple de topologie.

Chaque entrée de la table de routage possède le format suivant :  $[R\_dest, R\_next, R\_dist, R\_if\_d]$ . Cela signifie que le nœud identifié par  $R\_dest$  est joignable via le prochain saut dont l'identifiant est  $R\_next$ .  $R\_dist$  est la distance en nombre de sauts séparant  $R\_dest$  du nœud local et  $R\_if\_d$  est l'identifiant de l'interface locale par laquelle le nœud peut atteindre  $R\_dest$ . La figure 5.5 donne un exemple des tables de routage du réseau ad hoc de la figure 5.2.

#### 5.2.5 Raisons de choix du protocole OLSR

Une partie de cette thèse présente un protocole de routage offrant de la qualité de service. Il est basé sur le protocole OLSR. Le choix d'OLSR devient évident en regardant l'ensemble des avantages qu'OLSR peut offrir en tant que protocole de routage dans les réseaux ad hoc et en terme des idées d'optimisation intégrées et qui étaient reprises par d'autres protocoles.

OLSR est le résultat de six années de travail d'HIPERCOM, équipe de recherche de l'INRIA Recquencourt, qui a été retenu récemment par l'IETF comme un RFC [73]. OLSR se rapproche du

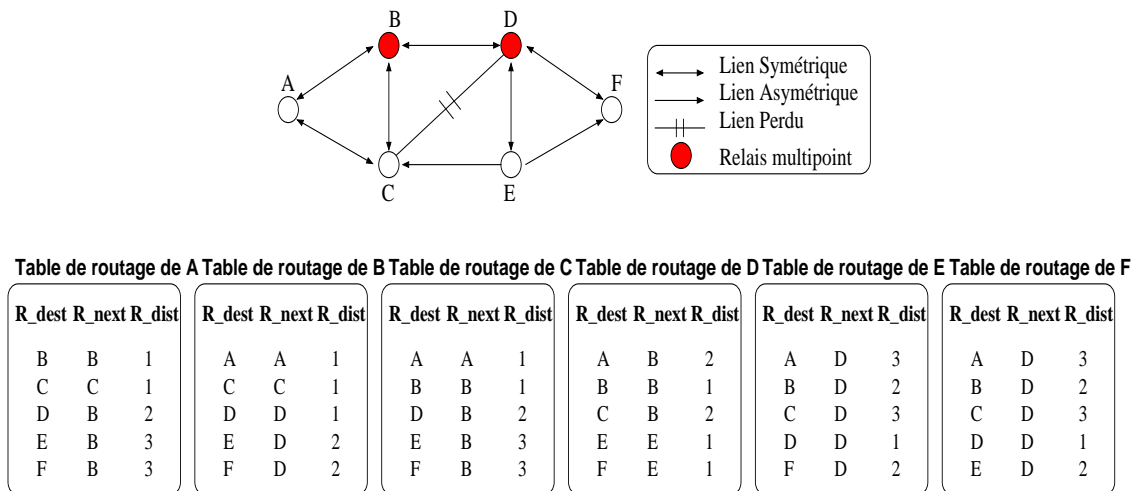


FIG. 5.5 – Exemple de routage maintenue par OLSR.

protocole OSPF (Open Shortest Path First), tous deux sont des protocoles de type état des liens où chaque nœud diffuse son voisinage dans le réseau. Mais OLSR est tout à fait adapté aux réseaux ad hoc : il peut contrôler les liens entre les équipements par des paquets spéciaux, les *Hellos*, et il est optimisé pour la diffusion. Cette optimisation économise une grande partie de la bande passante du réseau, ce qui est très important dans des réseaux denses. Elle s'appuie sur le concept de relais multipoint (MPR Multipoint Relay). OLSR présente aussi l'avantage de s'adapter parfaitement aux protocoles de l'Internet et il autorise chaque équipement à connaître la topologie du réseau à tout instant.

Les idées développées dans OLSR sont reprises dans d'autres travaux de l'IETF notamment dans le groupe OSPF. Plusieurs propositions de standard émanant des plus grandes compagnies américaines incluent l'idée des MPRs d'OLSR.

### 5.3 Définition de la qualité de service

QoS est un terme largement utilisé ces dernières années dans le domaine des réseaux filaires, mais en vérité il y'a beaucoup de débats sur sa signification exacte. La plupart des fournisseurs mettent en œuvre leurs protocoles de QoS sous des scénarios ou des hypothèses bien spécifiés comme la topologie du réseau, le modèle de mobilité, le modèle de trafic, etc. L'implémentation résultante n'est pas générale et elle peut ne pas fonctionner pour certains scénarios. Dans les recommandations E.800, le CCITT (United Nations Consultative Committee for International Telephony and Telegraphy) a défini la QoS comme : "Ensemble des effets portant sur les performances d'un service de communication et qui détermine le degré de satisfaction d'un utilisateur de ce même service". Cette définition est la plus largement acceptée puisqu'elle ne référence aucune métrique comme la bande passante, le délai, etc, ou un mécanisme comme le contrôle d'admission, SLA, protocole de signalisation, etc.

## 5.4 Les concepts de QoS sur IP

La Qualité de service dans les réseaux filaires peut être garantie par deux mécanismes : le sur-dimensionnement des ressources et l'ingénierie de trafic [88]. Avec le sur-dimensionnement, la bande passante du réseau est augmentée afin d'éviter la congestion, le délai des files d'attente, la perte des paquets, etc. Ceci permet au réseau de mieux supporter les applications multimédias. Cette augmentation de capacité est due aux améliorations des ressources physiques comme les liens (par fibre optique), routeurs, cartes réseau, etc. Cette approche est facile à mettre en œuvre puisque l'amélioration est faite graduellement, mais malheureusement il ne contient qu'une seule classe de service et tous les utilisateurs ont la même priorité. Le fonctionnement du réseau devient immaîtrisable lorsque le trafic est non uniforme où des parties du réseau subissent une concentration importante de trafic. L'idée principale de l'ingénierie de trafic est de pallier aux inconvénients de l'approche précédente par une classification des utilisateurs (ou leurs applications) en classes de service avec une priorité différente pour chaque classe. Cette approche a deux mécanismes qui sont complémentaires : i) ingénierie avec réservation : utilisée dans ATM (Asynchronous Transfer Mode) et aujourd'hui dans RSVP-IntServ [89–91], et ii) ingénierie sans réservation : utilisé actuellement dans DiffServ (Differentiated Services).

### 5.4.1 IntServ et DiffServ

Ces architectures de QoS ont été standardisées au sein de l'IETF (Internet Engineering Task Force) [4].

#### IntServ

Les Integrated Services (IntServ) vont permettre de gérer des flux entiers de données. Ce modèle propose deux classes de service en plus du *Best-Effort* :

- Le *Guaranteed Service* proposé aux applications ayant des contraintes sur les délais.
- Le *Controlled Load Service* proposé aux applications requérant un service Best-effort amélioré, plus fiable.

Deux points sont importants à propos d'IntServ. Premièrement, la philosophie IntServ consiste à réserver des ressources sur la base de micro-flux (c'est-à-dire, une connexion TCP, un flux UDP, ...). Deuxièmement, IntServ offre des garanties dures de bout-en-bout. Les routeurs implémentant ce modèle sont capables de séparer les micro-flux pour les traiter individuellement. Il apparaît tout de suite que des mécanismes complexes vont être nécessaires pour isoler et traiter chaque micro-flux.

Ces deux points impliquent l'utilisation de plusieurs composants : la signalisation, le contrôle d'admission, la classification des flux et l'ordonnancement des paquets. La signalisation permettant la réservation de ressources est assurée par le protocole RSVP (Ressource ReSerVation Protocol). D'autres protocoles de signalisation sont possibles, mais en pratique RSVP est le seul qui ait été déployé pour les services intégrés.

L'objectif du contrôle d'admission est de bloquer les micro-flux dont les ressources demandées ne sont pas disponibles. Ce contrôle est opéré par chaque routeur sur le chemin. Chacun d'entre eux accepte ou rejette la demande de service suivant l'état actuel du réseau. Les routeurs indiquent à l'application, via RSVP, si le besoin de QoS peut être satisfait ou non. Ensuite, la classification des micro-flux est, elle aussi, effectuée par chaque routeur. Cette phase complexe permet de séparer

les flux et d'insérer les paquets entrant dans les files d'attente appropriées. Enfin, l'ordonnanceur gère l'ordre de sortie des paquets des files d'attente afin d'assurer la QoS demandée.

On voit rapidement le défaut majeur de la philosophie IntServ. Pour un routeur central, le nombre de micro-flux est colossal. Pour chacun, il faut maintenir une file d'attente. Pour chaque paquet, il faut déterminer dans quelle file il doit être placé (et ce sur base de cinq champs contenus dans les en-têtes de transport et IP, à savoir adresse de destination, port de destination, numéro de protocole, adresse et port source). Et quand un nouveau micro-flux se présente, il faut tout réorganiser.

### DiffServ

La philosophie DiffServ tente quant à elle d'offrir une architecture qui présente une meilleure résistance au facteur d'échelle. DiffServ définit plusieurs catégories de services (de best-effort aux services premium). Les réservations ne se font plus par micro-flux mais bien par catégorie de services.

Il importe donc à chaque nœud d'attribuer pour chaque catégorie de services une file d'attente, et d'ordonnancer le tout pour satisfaire une certaine politique. Il faut bien entendu toujours classer le paquet entrant pour savoir dans quelle file d'attente il doit aller, mais cela est immédiat. Dans IPv4, c'est le champ ToS qui est utilisé à cette fin et le champ classe de trafic dans IPv6. L'ordonnanceur peut par ailleurs être configuré statiquement.

Deux types de routeurs sont donc définis :

- Les routeurs de bord (Edge Routers), chargés de la classification, du marquage et du maintien de l'état des flux.
- Les routeurs de cœur (Core Routers), chargés uniquement de l'acheminement des paquets selon le marquage.

La simplicité de DiffServ a un prix. À l'intérieur d'une catégorie de services, il n'y a pas de protection entre les flux. Il n'y a pas non plus d'admission au sens strict (comme il y en avait dans IntServ), donc DiffServ ne peut pas fournir des garanties dures.

### 5.4.2 RSVP

RSVP est à la base un protocole de réservation de ressources. Nous avons notamment déjà cité son usage dans le cadre d'une philosophie IntServ pour signaler des réservations. La principale motivation à l'origine de sa création était le support efficace d'applications multipoint multicast avec réservations de ressources (où il peut donc y avoir plusieurs émetteurs et plusieurs récepteurs). Remarquons que RSVP n'est pas concerné par le routage, qui est réalisé par un autre protocole de contrôle. RSVP supporte trois modes de réservations :

- FF (Fixed Filter) pour recevoir d'un seul émetteur.
- SE (Shared Explicit) pour recevoir d'un sous-ensemble d'émetteurs, qui typiquement n'émettent pas simultanément (les réservations étant agrégées sur les liens communs).
- WE (Wildcard Filter) pour recevoir de tous les émetteurs.

Les messages RSVP sont de quatre types : messages PATH, messages RESV, messages de tear-down et messages d'erreurs.

La source envoie des messages PATH aux récepteurs d'un groupe multicast, où est décrite la forme du trafic qu'elle émettra. Ces messages PATH suivent des chemins dictés par le protocole de routage utilisé. Un récepteur peut envoyer un message RESV qui effectue la réservation. Le message RESV contient la quantité de ressources que le récepteur veut réserver. Il suit (en sens

inverse) la même route que le message PATH, grâce à l'état qui a été mis en place par le message PATH. La réservation se fait nœud par nœud. Le message RESV ne remonte pas nécessairement jusqu'à la source (il ne remonte jusqu'à la source que si la demande de réservation est plus grande que la réservation déjà en place pour le groupe multicast). Les RFCs [89–91] constituent d'excellentes références.

### 5.4.3 Routage contraint

Un protocole de routage IP classique (comme OSPF par exemple) vise à trouver un chemin qui optimise une certaine métrique. Le routage IP avec contraintes ou CR (*Constraint-based Routing*) vise quand à lui à trouver un chemin qui optimise une certaine métrique tout en satisfaisant certaines contraintes (spécifications de performance, contraintes administratives, contraintes de ressources, ...).

Le routage avec contraintes implique de calculer le chemin à la source. En effet, seul le routeur ingress a connaissance des contraintes à respecter. Mais pour pouvoir calculer le chemin, il faut disposer non seulement de la topologie complète du réseau mais aussi de toutes les particularités des différents liens (en termes de bande passante utilisée, de spécifications administratives, ...). Un protocole de routage à état de liens s'impose donc pour disposer de la topologie complète. Et comme on le voit sur la figure 5.6, cela ne pose guère de problèmes de rajouter quantité d'informations spécifiques à un lien dans les paquets à état de liens d'OSPF. On trouvera plus d'informations sur les extensions d'OSPF liées à l'ingénierie de trafic dans le draft [92].

S			
A	d1	125.3	60
B	d2	22.9	200
C	d3	1024	40
D	d4	523.06	55.6
E	d5	2.6	350
F	d6	56	160

↓  
Voisins

↓  
Distance

↓  
**Bw**

↓  
**Del**

FIG. 5.6 – Données opaques.

Il importe ensuite d'utiliser un algorithme modifié pour le calcul du chemin avec contraintes. L'algorithme de Dijkstra qui est l'algorithme de calcul de OSPF peut être modifié pour prendre en compte des contraintes. L'algorithme modifié porte le nom de CSPF.

Enfin, il faut réserver des ressources tout le long du chemin. La réservation des ressources altérera les attributs des liens traversés et il faudra donc pouvoir rediffuser ces nouveaux états de liens par le protocole de routage. La mise en place d'une telle architecture peut être facilement réalisée grâce au duo RSVP/MPLS. MPLS (*Multiprotocol Label Switching*) [93] fournit une fonctionnalité de routage explicite très efficace.



## 5.5 Les modèles de QoS pour MANETs

Le modèle de QoS spécifie l'architecture qui offre des services meilleurs que ceux dans le modèle best-effort existant dans MANETs. Cette architecture doit prendre en compte les contraintes imposées par les réseaux ad hoc comme la topologie dynamique, capacité, etc. Nous avons décrit le concept de base des modèles de QoS pour le réseau Internet (IntServ/RSVP et DiffServ). Nous allons analyser les raisons pour lesquelles les modèles développés pour les réseaux filaires ne sont pas directement adaptés pour MANETs et ensuite nous introduisons les modèles existants proposés pour MANETs.

L'application du modèle IntServ/RSVP dans MANET s'avère inadaptée à l'environnement ad hoc due à la limite des ressources dans MANETs. Il existe plusieurs facteurs qui empêchent l'utilisation de ce modèle dans MANETs : i) chaque nœud doit se charger de gérer le contrôle d'accès, la classification et l'ordonnancement des flux. C'est une charge trop importante pour des terminaux aux ressources limitées ; ii) Les informations sur les flux augmentent proportionnellement au nombre de flux gérés. Il n'y a pas d'agrégation. Ce problème de passage à l'échelle n'est pas spécifique au MANETs, on le retrouve aussi dans l'Internet. La maintenance de cette quantité d'information par des terminaux mobiles dont les ressources sont limitées n'est pas envisageable. Même si aujourd'hui les MANETs restent de petite taille et ne sont destinés à gérer qu'un nombre restreint de flux, cette solution n'est pas fiable à long terme, puisqu'il est probable que les MANETs vont être amenés à se développer ; iii) Les paquets de signalisation RSVP utilisent une quantité non négligeable de bande passante sur des liens déjà limités.

Le modèle DiffServ quant à lui est simple puisque l'acheminement par les routeurs de cœur est très simple et rapide et donc sa complexité se trouve dans les nœuds de bord. Ce modèle semble le mieux adapté aux réseaux MANETs. Pour résoudre le problème de passage à l'échelle, ce modèle utilise une granularité par classe, où aucune signalisation pour la réservation de ressources n'est utilisée. Cependant, la question de la définition des routeurs de bord et des routeurs de cœur reste très *ambiguë* dans ces réseaux. Intuitivement, la source fait parti des routeurs de bords et les nœuds du chemin font parti des routeurs de cœur. Mais dans les MANETs, chaque nœud doit pouvoir jouer les deux rôles à la fois (figure 5.7), puisqu'il peut être source d'une communication et relais pour une autre. Cela engendrerait donc une charge trop importante sur tous les nœuds du réseau.

### 5.5.1 FQMM : Flexible quality of service model for MANETs

Le modèle FQMM [94] est le premier modèle de QoS proposé pour MANETs en 2000 par Xiao et al. Il repose sur une architecture réseau plate (non hiérarchique), constituée d'une cinquantaine de nœuds mobiles, formant un domaine DiffServ. Il combine les propriétés des modèles filaires IntServ et DiffServ, en offrant une méthode d'approvisionnement hybride : par flux, pour les trafics prioritaires, et par classe pour les autres trafics. Ce modèle est basé sur l'hypothèse que tous les paquets dans le réseau ne demandent pas la priorité maximale sinon on aura exactement le modèle IntServ où l'approvisionnement est par flux. Dans le réseau, les nœuds peuvent avoir des rôles différents suivant les trafics existants : nœud d'entrée du trafic (ingress node), intermédiaire (core node) ou de sortie (egress node) (figure 5.7).

Les nœuds d'entrée permettent de marquer et classer les paquets, qui seront ensuite relayés par les nœuds intermédiaires suivant leurs PHB (Per Hop Behavior) [95], jusqu'à arriver au nœud destinataire. Ce modèle repose essentiellement sur la couche IP, où les fonctionnalités sont séparées en deux grands plans : le plan relayage de données et le plan contrôle et gestion (figure 5.8).

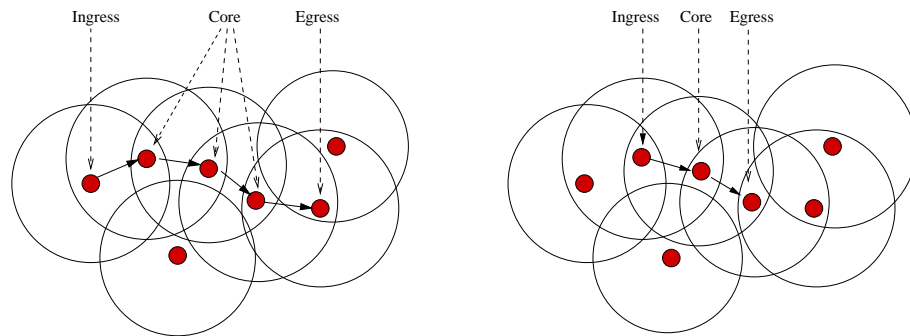


FIG. 5.7 – Le 1er scénario compte 3 nœuds intermédiaires tandis que le 2ème comporte un nœud intermédiaire.

Les techniques d'ordonnancement et de gestion de mémoires tampons sont étudiées. Dans ce modèle, le protocole de routage est supposé fournir des routes ayant suffisamment de ressources.

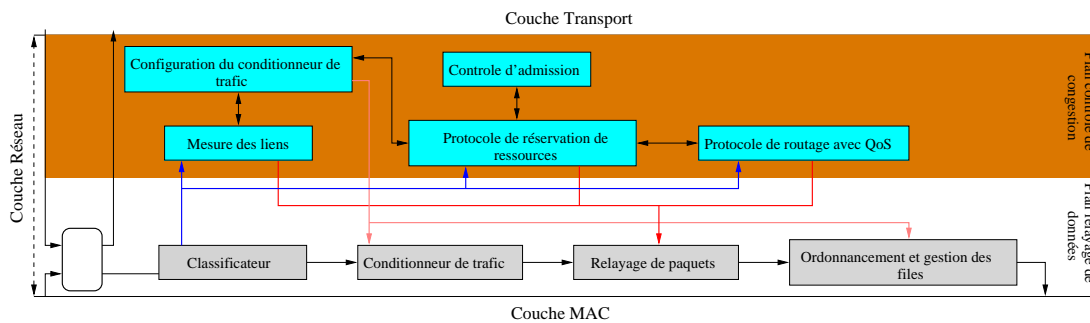


FIG. 5.8 – Le modèle FQMM.

L'avantage d'une telle approche est la possibilité d'interfacer le réseau avec l'Internet, vu les mécanismes de qualité de services offerts qui sont proches des protocoles filaires. Cependant, plusieurs mécanismes ainsi que l'interaction avec la couche MAC restent à définir pour s'adapter aux conditions variables du réseau ad hoc.

### 5.5.2 SWAN : Service differentiation in wireless ad hoc networks

SWAN [96] est un modèle réseau sans état basé sur des algorithmes de contrôle distribués dans le but d'assurer une différenciation de services dans les réseaux ad hoc (voir figure 5.9). Il offre la priorité (au niveau paquet) aux trafics temps réel en contrôlant la quantité de trafics best-effort acceptée par nœud. Pour accepter un nouveau trafic temps réel, le contrôle d'admission sonde la bande passante minimale disponible sur la route (valide et obtenu par un protocole de routage). Une décision à la source est alors prise suivant la bande passante obtenue. Pour maintenir la qualité de service des trafics déjà acceptés, le débit des trafics best effort est régulé en utilisant les mesures de délai au niveau MAC comme paramètre. Un classificateur et un shaper permettent de différencier les deux types de trafic. En cas de congestion, les bits ECN (Explicit Congestion Notification) de l'entête des paquets IP sont positionnés pour permettre à la source de re-initier le contrôle d'admission. Si la route ne dispose pas d'assez de bande passante, le trafic est supprimé. Ainsi, SWAN permet de fournir une QoS logicielle (soft QoS).

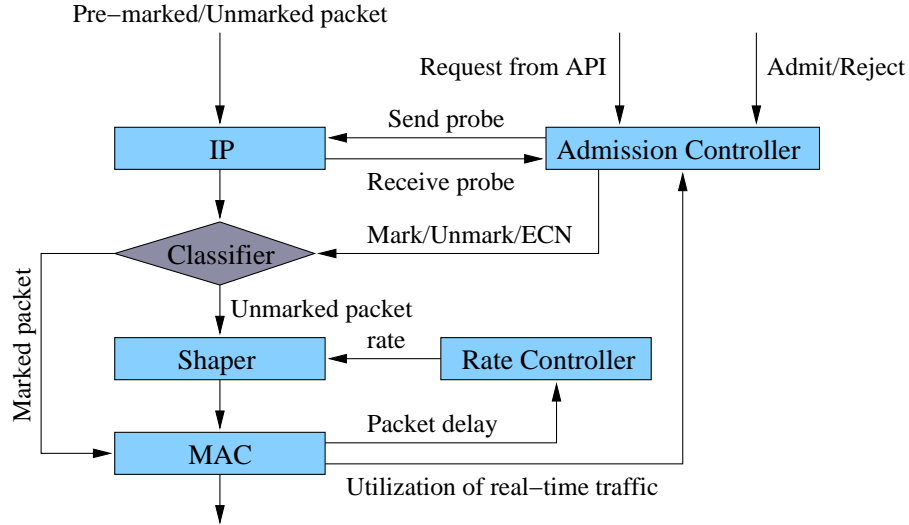


FIG. 5.9 – Le modèle SWAN.

Un flux prioritaire admis n'est pas sûr d'avoir des garanties pour l'entière durée de la communication, et peut à tout moment être violé par d'autres demandes de trafics. Un mécanisme de contrôle de débit des flux best effort n'est pas à lui seul suffisant pour offrir des garanties aux applications temps réel. En outre, dans cette approche, le protocole de routage ainsi que la couche d'accès au médium sont de type best-effort.

### 5.5.3 Modèle iMAQ

Le modèle iMAQ (*an Integrated Mobile Ad hoc Qos framework*) [97] fournit le support des transmissions des données multimédia dans MANETs. Le modèle inclut une couche ad hoc de routage et une couche de service logiciel (Middleware). Dans chaque nœud, ces deux couches partagent les informations et communiquent afin de fournir les garanties de QoS aux trafics multimédias. Le protocole de routage est basé sur la prédiction de la position des nœuds (predictive location-based) et orienté QoS. La couche Middleware communique également avec la couche application et la couche réseau et essaye de prévoir le partitionnement du réseau. Pour fournir une meilleure accessibilité aux données, il réplique les données entre les différents groupes du réseau avant d'effectuer le partitionnement.

## 5.6 Systèmes de signalisation pour la QoS dans MANETs

La signalisation pour la QoS sert à réserver et libérer les ressources dans le réseau. Il y a deux impératifs pour parvenir à une signalisation efficace : i) le transfert des signaux entre les routeurs doit être fiable, ii) ces signaux doivent être correctement interprétés afin de déclencher les mécanismes adéquats.

### 5.6.1 Signalisation in-band et out-of-band

On peut distinguer deux types de signalisation :

- *in-band* : les informations de contrôle de flux sont véhiculées avec les paquets de données.

- *out-of-band* : on utilise des paquets de contrôle spécifiques.

La signalisation in-band est plus *légère* que la signalisation out-of-band. En effet, le coût engendré par les messages out-of-band spécifiques à la signalisation est très important et consomme de la bande passante supplémentaire. De plus les paquets de signalisation doivent avoir une priorité supérieure à celle des paquets de données afin de garantir la QoS à tout moment. Cela mène à un système complexe et dont les performances vont devenir assez faibles. Mais cette approche (out-of-band) supporte le passage à l'échelle puisque les messages de contrôle ne dépendent pas de la transmission de paquets de données. RSVP est un exemple de ces protocoles de signalisation out-of-band. Les contraintes en ressources des MANETs ne permettent pas la mise en place d'un mécanisme complexe. Le but est d'offrir un protocole aussi léger et simple que possible. Mais comme RSVP n'a pas été mis au point en prenant en compte les contraintes des MANETs, il n'est pas efficace dans ce domaine.

### 5.6.2 Maintient des réservations soft-state et hard-state

On distingue deux méthodes pour le maintien des réservations dans le réseau. D'abord la méthode soft-state où les ressources réservées sont libérées si elles ne sont pas utilisées pendant un certain laps de temps. Par opposition, il existe aussi des méthodes efficaces et plus simple, appelées hard-state où les ressources ne sont libérées que lorsque cela est explicitement demandé. Dans ce deuxième cas, il n'y a pas besoin de signalisation ni de gestion de temporisateurs.

La solution soft-state est plus adaptée aux MANETs car les liaisons ne sont pas fiables et sont susceptibles d'être cassées. Dans certain cas où un nœud se retrouve isolé de la source pour laquelle il avait réservé des ressources, ce nœud ne recevra jamais de message de libération et réservera donc des ressources jamais utilisées. Il est donc important que les ressources soient libérées automatiquement si elles ne sont pas utilisées.

### 5.6.3 Protocole de signalisation INSIGNIA

INSIGNIA [98] fût le premier système de signalisation in-band permettant la QoS spécifiquement conçue pour les MANETs en 1998. L'information de signalisation est incluse dans les options des paquets IP (appelé option INSIGNIA), la signalisation est donc de type in-band. Ce système permet une gestion des services par flux, comme dans RSVP. De plus INSIGNIA utilise un système de réservation soft-state. Nous avons vu à la section précédente que c'était un bon choix dans des environnements dynamiques comme les MANETs. INSIGNIA utilise des algorithmes de réservation, restauration et adaptation dédiés aux MANETs. Ce protocole maintient les informations sur l'état des flux de bout-en-bout en informant les sources de l'état de leur flux.

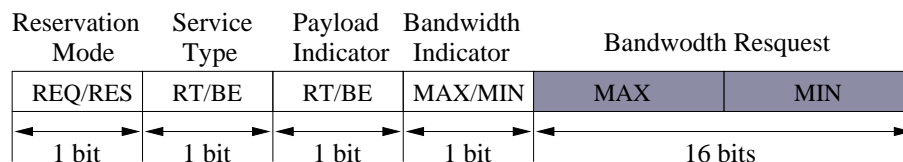


FIG. 5.10 – Les champs de l'option d'INSIGNIA.

Le champs de l'option d'INSIGNIA a une longueur de 20 bits (figure 5.10). Le bit "Reservation Mode" indique si ce paquet est en cours de réservation (REQ) ou s'il a déjà réservé des ressources (RES). Dans le cas REQ, le paquet est envoyé au module INSIGNIA qui va se charger

de la suite du traitement. Le module INSIGNIA va alors définir si des ressources peuvent être allouées à ce paquet ou non. Si les ressources peuvent être allouées, le champs “Service Type” est mis à RT (Real Time), sinon, il est descendu à BE (Best-Effort). Dans les deux cas, le paquet sera transmis au prochain nœud. Arrivé à la destination, le paquet contient donc soit la valeur RT soit la valeur BE. La quantité de bande passante demandée par le paquet se trouve dans le champ “Bandwidth Request”, qui indique un minimum et un maximum pour cette valeur. En se basant sur cette information, le module INSIGNIA peut déterminer la quantité de bande passante à attribuer. Enfin le bit “Bandwidth Indicator” est un drapeau utilisé par le récepteur pour savoir si la demande de bande passante maximale a été satisfaite. Les simulations montrent que ce mode de réservation est très rapide.

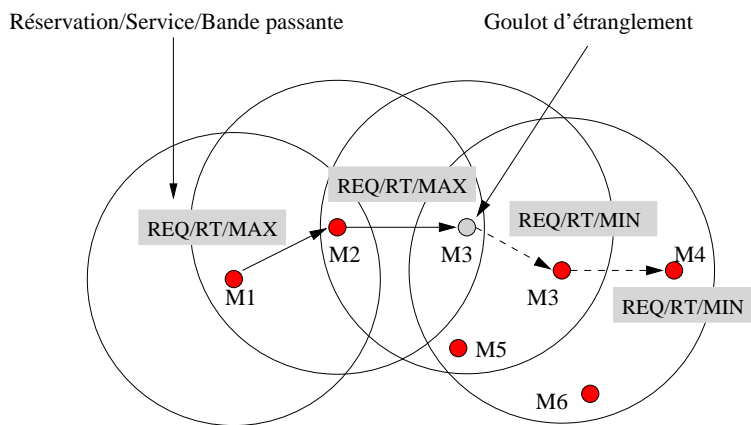


FIG. 5.11 – un nœud goulot d'étranglement.

Il faut noter que dans certaines situations, des nœuds peuvent être des goulots d'étranglement pour diverses raisons. Cela signifie que tous les flux qu'ils vont transmettre vont être dégradés. Par exemple (figure 5.11), un flux qui arrive avec le service RT et la bande passante maximale demandée peut être transmis par ce type de nœud en RT avec bande passante minimale ou même en BE. Il est important de rappeler que INSIGNIA est seulement un protocole de signalisation. On doit l'associer à un protocole de routage tel que AODV ou DSR qui va détecter les changements de topologie et mettre à jour les tables de routage. On a aussi besoin d'un module de contrôle d'admission qui va allouer les ressources après avoir déterminé si ces ressources sont disponibles. Finalement INSIGNIA est un protocole de signalisation efficace et bien adapté aux MANETs, principalement parce qu'il allie les avantages de la signalisation in-band et ceux de la réservation soft-state que l'on sait être très adaptés aux MANETs. INSIGNIA possède néanmoins quelques lacunes :

- D'abord un problème de passage à l'échelle. En effet, INSIGNIA suivant le modèle Intserv, le problème du maintien de l'état des flux dans chaque nœud ne peut être évité.
- Ensuite, la gestion de la bande passante n'est pas optimale puisque la réservation effectuée avant un goulot d'étranglement est perdue. Ce gaspillage n'a lieu que pendant un temps limité puisque la source va rapidement être informée du problème mais ce problème existe bel et bien.
- Enfin, INSIGNIA ne propose que deux classes de service : RT ou BE, une granularité plus fine pourrait permettre plus de flexibilité pour les applications.

#### 5.6.4 Le protocole Bruit

BRuIT (*Bandwidth Reservation under InTerferences influence*) [99] a pour but d'apporter un contrôle de la bande passante afin d'empêcher au maximum l'apparition de congestion dans le réseau et de fournir la bande passante demandée pour certains types de flux. Pour cela, il considère deux types de flux : les flux best-effort qui n'auront aucune garantie sur leur débit et les flux privilégiés à qui on peut réserver une certaine bande passante. Pour effectuer ces réservations et ce contrôle, BRuIT tente d'apporter suffisamment de connaissance à chaque mobile sur la bande passante qui est utilisée dans son voisinage étendu (l'ensemble de ses voisins à un et deux sauts) et ce régulièrement. Avec cette connaissance, il peut estimer la bande passante utilisée pour les flux privilégiés dans son voisinage étendu. À partir de là, chaque mobile peut décider de l'admission ou du rejet d'un flux privilégié qui nécessite une certaine bande passante. Ceci permet donc aux mobiles de n'accepter que les flux dont ils seront initialement en mesure d'honorer leur débit et donc d'empêcher, très souvent, l'apparition de congestion. BRuIT effectue une réservation de bande passante en se basant sur un protocole de routage avec qualité de service.

BRuIT ne permet pas de prendre en compte les flux transitant sur des mobiles en zone de détection de porteuse de certains autres mobiles, mais non connectés à ceux-ci par au plus deux sauts radio. Il reste encore du travail concernant l'utilisation de la bande passante, afin d'allouer une partie adaptative au trafic best effort fonction de la topologie et de l'utilisation du réseau

### 5.7 Routage avec QoS dans MANETs

Le routage avec QoS est un élément clé pour réaliser une architecture de QoS pour les MANETs. Le protocole de routage peut informer une source sur les conditions QoS du réseau. Cette connaissance va permettre l'établissement de connexions avec qualité de service. Il existe de nombreux protocoles de ce type.

Avant d'aller plus loin, il est important de définir les objectifs visés par le routage avec QoS. Le routage au mieux (sans QoS) consiste en général à trouver le plus court chemin en terme de distance ou de délais entre une source et une destination. Dans le cas du routage avec QoS, l'objectif n'est pas seulement de trouver le meilleur chemin selon un critère précis, mais de trouver un chemin "admissible". On ajoute donc des contraintes sur les routes pour déterminer si elles sont admissibles ou non. Par exemple, certaines applications peuvent nécessiter une certaine quantité de bande passante, un contrôle sur le taux de pertes, le délai, la gigue, etc. Toute route assurant un de ces critères peut être considérée comme assurant une certaine QoS. On voit apparaître des spécificités. Les routes doivent être calculées par flux et non par destination. En effet, un flux peut avoir des besoins de QoS alors qu'un autre flux entre ces mêmes nœuds en aura d'autres. Il est aussi indispensable de prendre en compte la fiabilité des liens dans le calcul des routes. Enfin, un protocole de routage ad hoc permettant la QoS doit pouvoir réagir très rapidement aux changements de topologie et les conditions QoS sans que les applications ne soient atteintes. Le but de ce type de protocole est donc de trouver une route dans le réseau qui puisse satisfaire de bout-en-bout les besoins en QoS demandés par une application. C'est une alliance entre un protocole de routage classique et un mécanisme de gestion des ressources. Le routage avec QoS est très difficile dans les MANETs car il engendre un overhead important. En effet, les nœuds doivent mettre en place un mécanisme permettant de stocker et mettre à jour les états de liens dans un environnement mobile. Cette mobilité rencontrée dans les MANETs rend le maintien de l'état précis des liens très difficile et très coûteux. De plus, la mobilité ou le manque d'énergie peuvent causer des ruptures dans les chemins établis, le protocole doit donc être capable de réagir très vite

à ce genre d'événement en recalculant des routes valides. L'idée est donc de trouver un équilibre entre le gain apporté par le routage QoS et l'importance de l'overhead. Plusieurs solutions ont été proposées pour MANETs.

### 5.7.1 CEDAR : Core-Extraction Distributed Ad hoc Routing Algorithm

Le protocole CEDAR [100] est un protocole de routage réactif qui semble bien réagir face au dynamisme des MANETs et fournit une qualité de service en terme de bande passante. CEDAR repose sur la notion de réseau de cœur. Ce réseau de cœur peut être vu comme un réseau overlay. Des nœuds sont choisis pour faire partie de cet overlay, ils forment alors le Dominating Set (DS). Deux cas peuvent se présenter pour chaque nœud du réseau : i) il fait partie de DS, c'est donc un nœud du cœur ; ii) l'un de ses voisins fait partie de DS.

Les nœuds de DS sont les représentants de leur zone. Chaque nœud n'appartenant pas au DS dépend d'un nœud de DS. Un chemin entre deux nœuds du cœur est appelé lien virtuel. L'un des objectifs principaux de ce protocole est de limiter au maximum l'inondation du réseau pour la découverte de route. CEDAR est divisé en trois composants : extraction d'un cœur du réseau, propagation d'état de lien et calcul de route.

Les tests effectués sur CEDAR montrent une efficacité assez intéressante, mais le fait que le routage soit fait à la source à la fois pour les paquets de contrôle et les paquets de données bride les performances de manière importante. C'est pourquoi, les travaux actuels sur ce protocole visent à le modifier en un protocole de routage de type prochain saut (next-hop routing). D'autre part, CEDAR ne sait gérer que la bande passante comme paramètre de QoS, ce qui peut être très insuffisant pour certaines applications ayant des contraintes de délai ou de gigue. Notons aussi qu'il n'est actuellement pas possible avec CEDAR d'offrir des garanties sur la bande passante disponible, la mobilité des nœuds étant imprévisible, seule une approximation est envisageable. Mais cette restriction paraît très difficile à surmonter car la mobilité des nœuds ne permet jamais de garantir la validité d'une route. CEDAR semble donc être une bonne solution pour des réseaux à faible mobilité.

### 5.7.2 Ticket-Based QoS Routing

Ticket-Based QoS Routing [101] est un protocole de routage distribué, qui autorise des informations d'état imprécises durant la phase de calcul de la route. Il permet de réduire la quantité des messages de routage diffusés pour la découverte de la route, en publiant un certain nombre de "tickets logiques". Chaque message de découverte (ou d'observation) de route doit avoir au moins un ticket. Quand un message arrive à un nœud, il peut être divisé en plusieurs messages d'observation, qui sont relayés vers les prochains sauts. Chaque message "fils" contiendra un sous ensemble des tickets de son message "père". Evidemment, un message ayant un seul ticket ne peut être divisé. Lors de l'arrivée d'un message de découverte de route à la destination, le chemin saut par saut est connu et les informations de délai ou de bande passante peuvent être utilisées pour effectuer la réservation de ressources pour la route répondant aux besoins de QoS. Le nombre de tickets générés est fonction de la précision des informations d'états disponibles à la source et les besoins de QoS de la communication. Plus de tickets sont publiés dans le but d'augmenter la chance de trouver un chemin désiré. Dans les réseaux filaires, une distribution de probabilité, selon des informations sur le délai ou la bande passante, peut être calculée. Cependant, cela reste inapproprié dans les réseaux ad hoc où les liens sans fil sont sujets à des cassures, où les informations d'états sont imprécises. Pour cela, un modèle simple a été proposé pour l'algorithme Ticket Based.

Il utilise l'historique et l'estimation des variations du délai, et une formule de lissage pour calculer le délai courant. Pour s'adapter aux changements de topologie, l'algorithme autorise différents niveaux de redondance de route. Il utilise aussi des techniques de réparation et de reroutage pour la maintenance des routes. La réparation des routes se fait en utilisant des reconstructions locales.

### 5.7.3 QoS pour AODV

Une extension du protocole de routage réactif AODV [78] pour le support de la qualité de service a été proposée dans [102]. Elle consiste à étendre les paquets RREQ et RREP durant la phase de découverte de la route. Un nœud recevant un message RREQ avec l'extension de QoS doit être capable de satisfaire la demande de service pour pouvoir rediffuser RREQ (s'il n'a pas une route à jour dans son cache) ou envoyer un message unicast RREP vers la source. Après l'établissement d'une route, si un nœud intermédiaire ne peut pas maintenir la demande de QoS exigée, un message ICMP QoS\_LOST sera initié et envoyé vers la source.

La table de routage de chaque nœud contient désormais 4 champs supplémentaires : le délai maximum, bande passante minimale, la liste des sources qui ont demandé des garanties de délai ou de bande passante.

Pour le délai, à chaque fois qu'un nœud intermédiaire reçoit un RREQ, il soustrait le délai indiqué dans ce message le `NODE_TRAVERSAL_TIME` (par défaut 40 ms), qui est le temps requis par ce nœud afin de traiter RREQ. Si le `NODE_TRAVERSAL_TIME` est supérieur au délai demandé, le message RREQ sera dropé. La figure 5.12 montre un exemple d'un tel processus. Le message RREQ2 est dropé par le nœud B puisque le délai demandé est inférieur au `NODE_TRAVERSAL_TIME` de B. La destination répond par un message unicast RREP et chaque nœud intermédiaire rajoute son `NODE_TRAVERSAL_TIME` dans le champs délai de ce message.

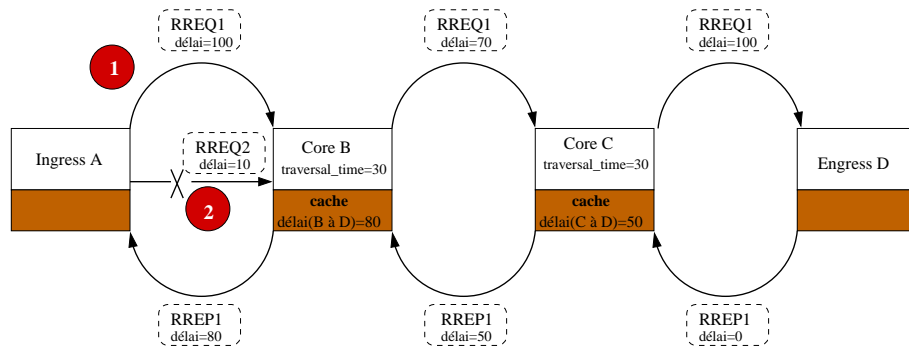


FIG. 5.12 – Le processus de relayage de RREQ selon le délai demandé.

Pour la bande passante, le champ correspondant dans les messages RREQ est préservé le long de la découverte de la route et chaque nœud intermédiaire rediffuse le message RREQ reçu si sa bande passante disponible est supérieure à la bande passante demandée. Une fois RREQ arrive à destination (ou un nœud intermédiaire ayant une route à jour vers la destination), cette dernière répond par un RREP qui se propage vers la source (chemin inverse) en inscrivant à chaque fois dans un nœud visité sa bande passante si elle est inférieure à la bande passante de RREP. La figure 5.13 montre le processus de traitement.

Le changement de `NODE_TRAVERSAL_TIME` ou la bande passante disponible d'un nœud cœur (intermédiaire) implique un changement des conditions de QoS sur tous les nœuds dépendants. Ce nœud envoie un message ICMP QOS\_LOSE à ses nœuds dépendants et c'est pour cette



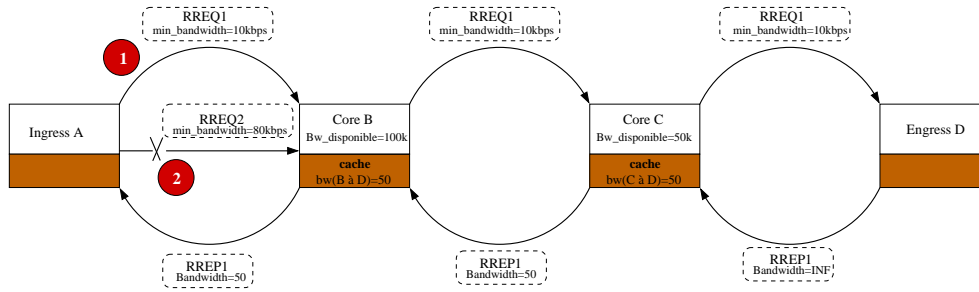


FIG. 5.13 – Le processus de relayage de RREQ selon la bande passante demandée.

raison que chaque nœud garde la liste des sources qui ont demandé des garanties de délai ou de bande passante.

L'inconvénient de l'approche à base des messages ICMP QOS\_LOST (il n'est pas mentionné dans le draft [102]) est qu'il n'y a aucune priorité de réservation ou prévention contre les violation des paramètres QoS requises.

## 5.8 Conclusion

Une grande partie des travaux dans les réseaux ad hoc se sont concentrés sur les problématiques de routage. L'avancée de ces travaux et l'augmentation des débits de communication des cartes sans fil permet d'envisager la mise en place d'applications nécessitant une certaine qualité de service. La qualité de service dans les réseaux ad hoc relève des défis qui n'ont pas été évoqués jusqu'à maintenant dans le cadre des réseaux filaires. La nature sans fil, la transmission en multi-saut et la mobilité rendent les modèles déjà développés pour les réseaux filaires IP inutilisables. Plusieurs travaux de qualité de service ont été proposés récemment pour les réseaux ad hoc qui peuvent être des modèles de QoS, signalisation et/ou routage avec QoS. Nous avons résumé les principaux travaux qui peuvent jouer un rôle important dans la définition d'un modèle de QoS complet et facile à mettre en œuvre.

## Chapitre 6

# EQMM : Efficient Quality of service Model for MANETs

En se basant sur les modèles de QoS et les protocoles de signalisation et de routage existants et avec une vision critique et innovatrice, nous proposons un modèle complet pour gérer la QoS dans MANETs. Ce modèle est appelé EQMM (*Efficient Quality of service Model for MANETs*). Il assure une QoS de bout-en-bout et réagit rapidement face aux changements topologiques et/ou conditions de QoS via un protocole de routage proactif avec QoS. Il limite l'apparition des congestions et la surutilisation de la bande passante par un mécanisme de contrôle d'admission suivi d'une réservation. Il possède aussi un mécanisme de contrôle de congestion qui consiste à faire disparaître les régions de congestions dues à la mobilité. EQMM est conçu pour être mis en œuvre réellement dans le cadre du projet SAFARI [6]. Par conséquent, les techniques qu'on propose doivent refléter la réalité et être facile à mettre en œuvre.

Le point de départ de ce chapitre consiste à décrire l'architecture de notre modèle. Ensuite, nous détaillons chacun de ses modules en exprimant l'interaction et le fonctionnement global. En fin, les performances du protocole QOLSR sont étudiées par simulation.

### 6.1 Description du modèle

Le modèle EQMM est basé sur des algorithmes distribués qui assurent une différenciation de services dans les réseaux ad hoc. Il distingue deux types de flux : best-effort et QoS. Un flux best-effort (E-mail, FTP) est un flux sans aucun paramètre spécifique (QoS, priorité), ni aucune garantie absolue contrairement à un flux QoS qu'il exige une route avec une garantie de QoS de bout-en-bout. Les flux QoS sont favorisés sur les flux best-effort et ils peuvent avoir plusieurs priorités. EQMM offre la priorité (au niveau paquet) aux trafics temps réel en contrôlant la quantité de trafics acceptée par nœud. L'architecture de notre modèle est illustrée dans les figures 6.1. Les flux best-effort et QoS traversent d'abord le module QOLSR qui est un protocole de routage avec QoS basé sur le protocole OLSR. QOLSR est considéré comme une première phase d'une procédure de contrôle d'admission puisque les flux QoS sont rejetés si aucun chemin n'est trouvé satisfaisant la demande. En cas de présence d'un chemin satisfaisant les contraintes d'un flux QoS, un contrôle d'admission est nécessaire pour vérifier la validité de ce chemin et réserver les ressources sur chacun de ses nœuds intermédiaires. Le contrôle de congestion consiste en deux algorithmes : la gestion des files et l'ordonnancement. Le premier gère la taille des files en éliminant des paquets si nécessaire tandis que le second détermine quel est le prochain paquet à envoyer sur le lien.

La protection du flux QoS prioritaire par rapport au flux best-effort passe obligatoirement par un bon dimensionnement des paramètres de configuration des mécanismes d'ordonnancement et de gestion des files d'attente. La composante d'estimation des paramètres de QoS mesure les valeurs des métriques (la bande passante disponible, le délai, la gigue, la perte des paquets, etc) sur les liens vers les voisins et communique ces informations au protocole QOLSR et aux modules de contrôle d'admission et congestion.

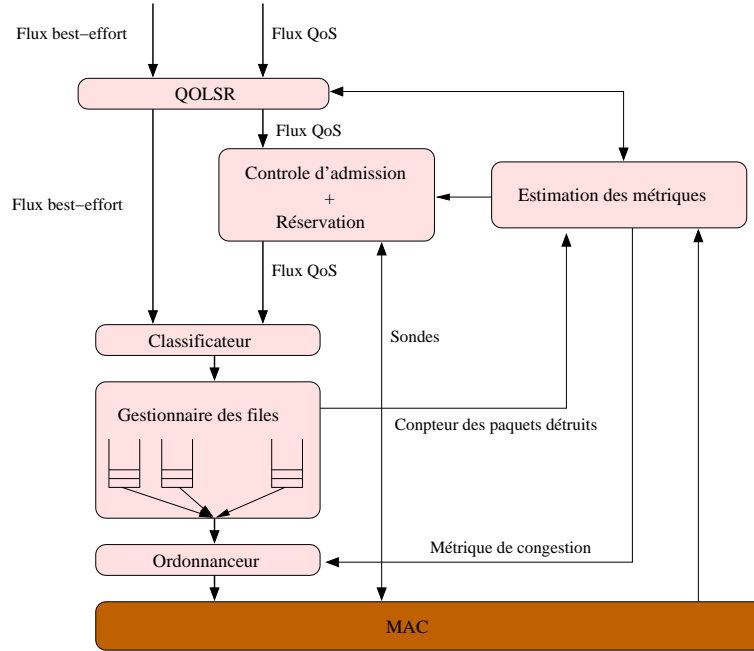


FIG. 6.1 – Le modèle EQMM.

La tâche d'un nœud intermédiaire consiste à : i) effectuer un contrôle d'admission suivi d'une réservation sur la base des sondes envoyées par la source et la destination, ii) effectuer une classification selon une ou plusieurs valeurs contenues dans l'entête et iii) détecter et contrôler les congestions dans le réseau. Le contrôle de congestion par les nœuds intermédiaires permet d'épargner la source d'assumer la tâche toute seule. Si les points de congestions persistent, les sources des flux QoS informés de la congestion via les messages *TC* peuvent décider de retrouver d'autres chemins.

Cette architecture contient cinq composantes principales : 1) un protocole de routage avec QoS, 2) une estimation des métriques, 3) un contrôle d'admission et de réservation, 4) une classification du trafic et 5) un contrôle de congestion. Les modules QOLSR et estimation des métriques coopèrent étroitement pour mesurer les valeurs des métriques localement et les propager ensuite dans le réseau. Les messages de contrôle de QOLSR échangés dans le réseau servent aussi à calculer quelques métriques comme la bande passante et le délai. Le protocole QOLSR offre pour chaque type de flux une route optimale selon la demande. Pour chaque flux QoS, le nœud source et les nœuds intermédiaires dans le chemin calculé par QOLSR performant un contrôle d'admission et une réservation en se basant sur la connaissance de la bande passante dans le réseau. Dans MANET, malgré les procédures de contrôle d'admission et réservation, le réseau peut toujours développer des congestions suite à une mobilité ou changement de connectivité. Pour cette raison, un contrôle de congestion est extrêmement important dans notre architecture. Il surveille en

permanence l'utilisation de la bande passante et la perte des paquets de données.

## 6.2 OLSR : QoS implicite

Le protocole OLSR n'utilise aucun paramètre de QoS explicitement et il n'offre aucune garantie pour les flux QoS. C'est un protocole pour les flux best-effort sur la base d'une seule métrique : le nombre de sauts. Mais en analysant le fonctionnement du protocole OLSR, nous remarquons que la QoS n'est pas réellement ignorée. En effet, OLSR a mis en œuvre trois techniques :

- utilisation des MPRs : cette technique permet d'économiser la bande passante et les autres métriques corrélées.
- utilisation des plus courts chemins : un plus court chemin en termes de nombre de saut consomme moins de bande passante et minimise les interférences inter et intra-flux.
- utilisation des mécanismes de contrôle de la qualité des liens : afin d'éviter d'une part l'utilisation des liens fragiles et instables qui entraînent des pertes de paquets de données et d'autre part les changements de routes fréquents et inutiles, deux solutions ont été proposées [73] : hystérisis des liens et contrôle de la puissance du signal.

## 6.3 Le protocole de routage QOLSR

Le protocole QOLSR (*QoS for OLSR*) est une extension du protocole de routage OLSR pour offrir la QoS dans les réseaux MANETs. Il est capable de trouver pour chaque type d'application une route optimale selon ses contraintes exigées. Des champs additionnels pour la QoS sont rajoutés aux messages de contrôle (*Hello* et *TC*) déjà existants dans le protocole de base OLSR.

Le protocole QOLSR est proactif, et donc les nœuds dans le réseau doivent maintenir une carte topologique à jour avec les informations de QoS les plus récentes. Le maintien d'une carte topologique permet le calcul rapide et immédiat d'une route pour chaque flux. Chaque nœud mesure les métriques de QoS sur les liens vers chacun de ses voisins comme la bande passante disponible, le délai, la probabilité de perte, etc. Ces informations seront enregistrées dans la table de voisinage pour calculer la liste des MPRs. Chaque nœud MPR diffuse la topologie de son voisinage avec les paramètres de QoS dans le réseau entier et seuls ces nœuds ont le droit de retransmettre la topologie locale. QOLSR considère la bande passante et le délai comme des métriques obligatoires et les autres métriques comme optionnelles.

Les Messages *TC* permettent à chaque nœud d'apprendre la topologie du réseau. Cette topologie est partielle puisqu'un message *TC* ne diffuse que les liens reliant un MPR avec ses sélecteurs. Par conséquent, un nœud lointain n'aura aucune information sur les liens entre les nœuds non MPR même si les diffusions des *TC* sont fiables (sans perte). La figure 6.2 montre la topologie entière du réseau et la topologie partielle diffusée par QOLSR obtenue par le nœud 1.

La topologie partielle peut avoir lieu aussi suite aux pertes des messages *TC* dues aux interférences, disfonctionnement dans le protocole, etc, ce qui favorise une possible absence de certains nœuds dans la carte topologique. Il faut noter que les cartes topologiques construites par chaque nœud ne sont pas similaires mais elles convergent toujours vers la même carte topologique partielle. Malgré cette topologie partielle, QOLSR est conçu pour trouver toujours des chemins de même performance que ceux trouvés dans le réseau entier.

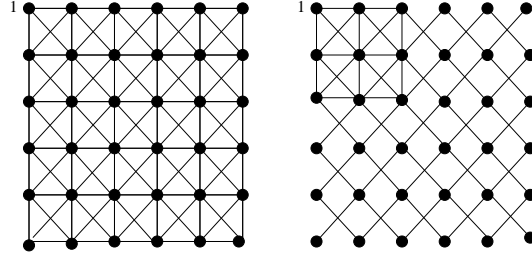


FIG. 6.2 – Topologie entiere et topologie partielle.

### 6.3.1 Principe de fonctionnement de QOLSR

Comme son nom l'indique, QOLSR est un protocole à état de lien optimisé ; il obtient des routes optimales selon les contraintes exigées par les flux. Alors que dans un protocole à état de lien, chaque nœud déclare ses liens directs avec ses voisins à tout le réseau, dans le cas de QOLSR, les nœuds ne déclarent qu'une sous-partie de leur voisinage avec leurs paramètres QoS grâce à la technique des relais multipoints. Malgré une connaissance pareille de la topologie de réseau, la source trouve toujours des routes qui ont les mêmes performances que celles dans le réseau entier. Ceci est assuré par une nouvelle heuristique pour le calcul des MPRs.

QOLSR est composé de plusieurs fonctions qui assurent la tâche de routage avec QoS :

- *Détection de voisinage* : chaque nœud diffuse un message *Hello* avec (TTL=1) pour détecter ses voisins ayant un lien direct et symétrique avec lui. Ce message contient les informations relatives aux voisins entendus par ce nœud.
- *Mesure des métriques* : chaque nœud doit estimer les paramètres de QoS sur les liens vers chacun de ses voisins. Ensuite, ces informations de QoS sont transmises localement en utilisant les champs de QoS des messages *Hello* (voir section 6.3.2). Par conséquent, chaque nœud aura une carte topologique pondérée à deux sauts avec des paramètres de QoS sur les liens.
- *Sélection des relais multipoint* : en se basant sur l'information de voisinage, chaque nœud sélectionne indépendamment sa liste des relais multipoints (MPRs). Contrairement aux MPRs classiques calculés dans OLSR, les MPRs d'un nœud dans QOLSR sont calculés pour contenir un sous-ensemble de ses voisins à un saut qui offrent une bande passante maximale et un délai minimum aux voisins à deux sauts. Les méthodes proposées pour la sélection des MPRs sont montrées dans la section 6.3.3. L'optimalité de cet ensemble n'est pas nécessaire mais par contre il est préférable que son cardinal soit de petite taille pour réduire le nombre de retransmissions redondantes dans la même région du réseau. Les nœuds sélectionnés comme MPRs sont ensuite déclarés dans les messages *Hello* pour atteindre les MPRs eux-mêmes. Les MPRs sont recalculés à chaque changement d'un lien symétrique (apparition, disparition ou changement des paramètres de QoS) vers un voisin à un saut ou à deux sauts.
- *Déclaration des MPRs sélecteurs et les métriques* : chaque nœud sélectionné comme MPR dans le réseau diffuse périodiquement à tous les nœuds du réseau des extensions des messages de topologie (*TC*) avec une fréquence déterminée par l'intervalle *TC\_Interval*. Un message *TC* généré par un nœud MPR *S* contient les nœuds ayant sélectionné le nœud *S* comme relais multipoint ainsi les valeurs des métriques sur les liens. Par ailleurs, un nœud

MPR peut envoyer des messages *TC* supplémentaires pour réagir plus rapidement aux changements de topologie et des informations de QoS. Uniquement les nœuds MPRs ont le droit de rediffuser les *TC*. La réception de ces messages, permet à chaque nœud dans le réseau de maintenir une base contenant les informations topologiques du réseau qui donne une vision globale (mais non pas entière) de la topologie du réseau et par conséquent les conditions de QoS.

- *Calcul des routes* : chaque nœud dans le réseau calcule sa table de routage en se basant sur les informations de voisinage et de topologie rafraîchies périodiquement. La table de routage est mise à jour selon les exigences d'une application ou de l'utilisateur. Pour satisfaire les contraintes de QoS demandées par un flux QoS, QOLSR fait appel à des algorithmes distribués correspondants aux types de problèmes à résoudre. Par exemple, l'algorithme RLMA pour deux métriques additives (voir chapitre 4).

### 6.3.2 La détection de voisinage et mesure des métriques

L'échange des messages *Hello* permet à chaque nœud d'accumuler des informations sur le réseau. Ces informations sont enregistrées dans la base d'information de voisinage.

#### La base d'information de voisinage

Cette base contient les ensembles des voisins à 1-saut, 2-sauts et sélecteurs des MPRs :

- *L'ensemble des voisins à 1-saut* : une entrée dans cet ensemble consiste en un tuple de voisins à 1-saut de type  $[N\_addr, N\_status, N\_Willingness, N\_bandwidth, N\_delay, N\_met\_1, \dots, N\_met\_n, N\_time]$ , où  $N\_addr$  représente l'adresse d'un voisin ;  $N\_status$  est l'état de lien avec ce voisin (MPR, Symétrique, Asymétrique, Perdu) ;  $N\_Willingness$  est un entier entre 0 et 7 pour exprimer la volonté de ce voisin à relayer des messages ;  $N\_bandwidth$ ,  $N\_delay$ ,  $N\_met\_1$ , ... et  $N\_met\_n$  désignent respectivement la bande passante disponible, le délai et les  $n$ -métriques additionnelles sur le lien avec ce voisin et enfin  $N\_time$  indique le temps d'expiration de cette entrée.
- *L'ensemble des voisins à 2-sauts* : comme la bande passante et le délai sont utilisés pour sélectionner des MPRs, seules ces deux métriques sont enregistrées. Une entrée dans cet ensemble consiste en  $[N\_addr, N\_2hop\_addr, N\_2hop\_bandwidth, N\_2hop\_delay, N\_time]$ , où  $N\_addr$  représente l'adresse du voisin à 1-saut ;  $N\_2hop\_addr$  l'adresse de voisins à 2-sauts ;  $N\_time$  est le temps d'expiration de cette entrée ;  $N\_2hop\_bandwidth$  et  $N\_2hop\_delay$  désignent la bande passante disponible et le délai sur le lien entre  $N\_addr$  et  $N\_2hop\_addr$ .
- *L'ensemble des sélecteurs des MPRs* : un nœud MPR doit maintenir la trace de ses voisins qui l'ont sélectionné comme MPR. Un tuple des sélecteurs des MPRs est présenté par  $[MS\_addr, MS\_bandwidth, MS\_delay, MS\_met\_1, \dots, MS\_met\_n, MS\_time]$ , où  $MS\_addr$  est l'adresse du voisin qui a sélectionné le nœud comme MPR ;  $MS\_bandwidth$ ,  $MS\_delay$ ,  $MS\_met\_1$ , ...,  $MS\_met\_n$  désignent respectivement la bande passante disponible, le délai et les  $n$ -métriques additionnelles sur le lien avec le nœud d'adresse  $MS\_addr$ .

### Le nouveau format d'un message *Hello*

Une extension est ajoutée aux messages *Hello* pour supporter la QoS dans MANETs. Un message *Hello* est diffusé localement à tous les voisins à 1-saut sans relaiage. Il contient :

- La liste des adresses des voisins avec lesquels il existe un lien symétrique et les valeurs de QoS sur chacun de ces liens.
- La liste des adresses des voisins qui ont été entendus auparavant (lien unidirectionnel) et les valeurs de QoS sur chacun de ces liens.
- La liste des voisins qui ont été sélectionnés comme MPRs et les valeurs de QoS sur les liens entre le sélecteur et ses MPRs.

Le nouveau format de message *Hello* est explicité en annexe A.1.

### Le traitement d'un message *Hello*

À la réception d'un message *Hello*, le nœud procède à la mise à jour de sa base d'information de voisinage. Le processus à suivre est le même que celui décrit dans [73] mais avec un traitement supplémentaire pour la QoS. En effet, chaque nœud doit mettre à jour les valeurs de la bande passante, le délai et les autres métriques dans l'ensemble des voisins à 1-saut, 2-sauts et les sélecteurs des MPRs.

#### 6.3.3 La sélection des relais multipoints (MPRs)

La sélection des MPRs consistent essentiellement, en un nœud donné, à ignorer un ensemble de liens et de voisins directs, qui sont redondants pour le calcul des routes de plus court chemin : plus précisément, dans l'ensemble des voisins d'un nœud, seul un sous-ensemble de ces voisins est considéré comme pertinent. Il est choisi de façon à pouvoir atteindre tout le voisinage à deux sauts (tous les voisins des voisins), cet ensemble est appelé l'ensemble des relais multipoints. Ces relais multipoints sont utilisés de deux façons : pour diminuer le trafic dû à la diffusion des messages de contrôle dans le réseau, et aussi pour diminuer le sous-ensemble des liens diffusés à tout le réseau puisque les routes sont construites à base des relais multipoint.

Choisir cet ensemble peut sembler facile à première vue. Mais en réalité, ce genre de problèmes revient à trouver un ensemble dominant qui appartient à la famille des problèmes NP-complet [103, 104]. Pour calculer cet ensemble, un nœud  $x$  doit acquérir la topologie locale à deux sauts (voisins à 1-saut et 2-sauts) et ensuite il utilise une heuristique distribuée.

Dans ce qui suit nous utilisons la terminologie suivante :

- $MPR(x)$  : l'ensemble des relais multipoint d'un nœud  $x$  ;
- $N(x)$  : l'ensemble des voisins à 1-saut ayant des liens symétriques avec  $x$  ;
- $N2(x)$  : l'ensemble des voisins à deux sauts ayant des liens symétriques avec au moins un nœud de  $N(x)$ .  $N2(x)$  ne contient aucun nœud appartenant aussi à  $N(x)$  ;
- $D(x, y)$  : le degré d'un nœud  $y \in N(x)$  est le nombre des voisins à 2-sauts couverts par celui-ci présent dans  $N2(x)$ .  $D(x, y)$  est donné par la formule suivante :  $D(x, y) = |N(y) - x - N(x)|$  où  $N(y) \subseteq N2(x)$  ;

- Le chemin le plus large : un chemin ayant la bande passante maximale calculée par un nœud source vers une destination en utilisant la topologie partielle diffusée par les messages *TC*. Dans ce chemin, chaque nœud intermédiaire est un MPR de son prédécesseur. Le chemin le plus large peut ne pas être unique.
- Le chemin le plus large de référence : un chemin ayant la bande passante maximale calculée par un nœud source vers une destination en utilisant la topologie entière du réseau ;
- shortest-widest chemin : c'est le plus court chemin en délai parmi les chemins les plus larges (qui ont la même bande passante maximale). Ce chemin est calculé par un nœud source vers une destination en utilisant la topologie partielle diffusée par les messages *TC* ;
- shortest-widest chemin de référence : c'est un shortest-widest chemin calculé entre deux nœuds en utilisant la topologie entière.

Une heuristique pour la sélection des MPRs a été proposée dans [105] et analysée dans [106]. Cette heuristique donne une solution optimale à  $\log n$  près avec  $n$  le nombre de voisins du nœud calculant l'ensemble des relais multipoints.

Avec la philosophie de sélection des MPR dans OLSR, des voisins à un saut avec des liens de meilleure qualité peuvent ne pas figurer parmi les nœuds MPRs et donc des routes de meilleures performances peuvent être négligées.

**Théorème 6.3.1** *Il n'y a aucune garantie qu'un nœud source trouve un chemin de même performance que le chemin le plus large de référence ou le shortest-widest chemin de référence vers une destination en utilisant l'heuristique de sélection des MPRs classique d'OLSR.*

PREUVE :

La démonstration de ce théorème est facile et elle peut être effectuée par un contre exemple à partir de la figure 6.3 et la table 6.1.

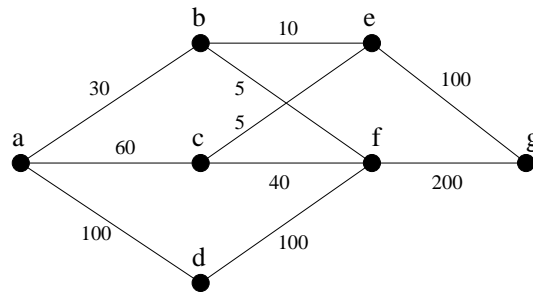


FIG. 6.3 – La topologie pondérée d'un contre exemple.

Lorsque le nœud *g* calcule sa route en utilisant le protocole OLSR pour rejoindre le nœud destinataire *a* il choisit le chemin  $g \rightarrow f \rightarrow b \rightarrow a$  avec 5 comme bande passante. Le chemin le plus large de référence entre *g* et *a* est  $g \rightarrow f \rightarrow d \rightarrow a$  avec 100 comme bande passante.



La manière de sélectionner les MPRs est importante pour déterminer les chemins optimaux en terme de bande passante et de délai même avec une topologie partielle. Dans la sélection des MPRs, un nœud voisin avec un lien de bande passante disponible importante et un faible délai doit être considéré comme une priorité après ceux qui connectent les voisins à deux sauts isolés (voisins à 2-sauts connectés par un seul lien avec l'un des voisins à 1-saut).



Nœud	Voisins à 1-saut	Voisins à 2-sauts	MPRs
a	b, c, d	e, f	b
b	a, e, f	c, d, g	f
c	a, e, f	b, d, g	f
d	a, f	b, c, g	f
e	b, c, g	a, f	b
f	b, c, d, g	a, e	c
g	e, f	b, c, d	f

TAB. 6.1 – Les MPRs sélectionnés en utilisant l'heuristique classique d'OLSR.

### 6.3.4 La sélection des MPRs-1

La première amélioration qu'on propose consiste à maintenir la même heuristique utilisée dans le protocole OLSR classique mais par contre lorsqu'il y'a plusieurs voisins à 1-saut qui couvrent le même nombre de voisins à 2-sauts non encore visités, on choisit celui qui offre un shortest-widest lien (un lien avec une bande passante maximale et en cas d'égalité avec d'autres liens, il possède un délai minimum). Les MPRs sélectionnés sont appelés MPRs-1. L'heuristique de sélection des MPRs-1 est la suivante :

**Étape 1 :** Commencer par un ensemble vide de MPRs-1 ;

**Étape 2 :** Choisir les nœuds de l'ensemble des voisins  $N(x)$  qui sont les seuls ayant un lien avec un nœud de  $N2(x)$ . Ajouter ces nœuds sélectionnés de  $N(x)$  à l'ensemble  $MPR-1(x)$ , et éliminer tous les nœuds à 2-sauts couverts par ces derniers de l'ensemble  $N2(x)$  ;

**Étape 3 :** Tant que  $N2(x) \neq \emptyset$

**Étape 3.a :** Pour chaque nœud  $y \in N(x)$  qui n'appartient pas à  $MPR-1(x)$ , calculer son degré  $D(x, y)$ .

**Étape 3.b :** Sélectionner le nœud de  $N(x)$  possédant le degré maximal comme un nœud MPR-1. S'il est unique, ajouter ce nœud sélectionné de  $N(x)$  à l'ensemble  $MPR-1(x)$  et éliminer tous les nœuds à 2-sauts couverts par ce MPR-1 ;

**Étape 3.c :** En cas d'égalité dans l'étape précédente (3.b), sélectionner comme MPR-1 le nœud qui à un shortest-widest lien avec  $x$ . Ajouter ce nœud sélectionné de  $N(x)$  à l'ensemble  $MPR-1(x)$  et éliminer tous les nœuds à 2-sauts couverts par ce MPR-1 ;

**Étape 4 :** Pour optimiser l'ensemble des MPRs-1, éliminer chaque nœud dans  $MPR(x)$ , un à chaque fois, et vérifier si toujours l'ensemble  $MP-1R(x)$  couvre tous les nœuds de  $N2(x)$ .

Le problème de choix entre plusieurs nœud qui peut se présenter à l'issue de l'étape 3.b est facile à résoudre dans cette heuristique. En effet, l'étape 3.c prend celui qui a un lien avec une bande passante maximale et délai minimum contrairement à l'heuristique d'OLSR qui sélectionne

un nœud d'une manière aléatoire. L'heuristique de la sélection des MPRs-1 a la même complexité que celle de la sélection des MPRs classique.

Le théorème suivant montre que l'utilisation des nœuds MPRs-1 dans le protocole QOLSR ne peut pas garantir des meilleurs performances que celle des MPRs.

**Théorème 6.3.2** *Il n'y a aucune garantie qu'un nœud source trouve un chemin de même performance que le chemin le plus large de référence ou le shortest-widest chemin de référence vers une destination en utilisant l'heuristique de sélection des MPRs-1 dans le protocole QOLSR.*

PREUVE :

On garde la figure 6.3 comme un contre exemple. En utilisant l'heuristique de sélection des MPRs-1, on aura la table 6.2 :

Nœud	Voisins à 1-saut	Voisins à 2-sauts	MPRs-1
a	b, c, d	e, f	c

TAB. 6.2 – Les MPRs-1 sélectionnés.

Lorsque le nœud  $a$  sélectionne ses MPRs-1, il aura le choix entre  $b$  et  $c$  à l'issue de l'étape 3.b. Puisque le lien  $(a, c)$  a plus de bande passante que le lien  $(a, b)$ , alors l'étape 3.c choisit le nœud  $c$  comme MPRs-1. La route calculée par le nœud  $g$  vers la destination  $a$  est  $g \rightarrow f \rightarrow c \rightarrow a$  avec une bande passante de 40 alors le chemin le plus large de référence entre  $g$  et  $a$  est  $g \rightarrow f \rightarrow d \rightarrow a$  avec 100 comme bande passante.



### 6.3.5 La sélection des MPRs-2

Les MPRs-2 d'un nœud  $x$  sont les voisins à 1-saut de  $x$  qui offrent pour chacun des voisins à 2-sauts de  $x$  une bande passante maximale et délai minimum. Cet ensemble permet à chaque nœud d'atteindre son voisinage à deux sauts avec les meilleurs paramètres de QoS possibles. L'algorithme de sélection des MPRs-2 est le suivant :

**Étape 1 :** Commencer par un ensemble vide de MPRs-2 ;

**Étape 2 :** Pour chaque nœud  $z$  de  $N2(x)$  :

**Étape 2.a :** Pour chaque nœud  $y \in N(x)$ , calculer son degré  $D(x, y)$ .

**Étape 2.b :** Sélectionner comme MPR-2 le nœud  $y \in N(x)$  vérifiant que  $(x, y, z)$  est shortest-widest chemin. Si  $y$  est unique, ajouter  $y$  à l'ensemble  $MPR-2(x)$  et éliminer le nœud  $z$  de l'ensemble  $N2(x)$  ;

**Étape 2.c :** En cas d'égalité dans l'étape précédente (2.b), sélectionner comme MPR-2 le nœud  $y$  possédant le degré maximal. Ajouter  $y$  à l'ensemble  $MPR-2(x)$  et éliminer le nœud  $z$  de l'ensemble  $N2(x)$  ;

Cet algorithme peut être facilement remplacé par un algorithme des shortest-widest chemins appliqué sur une topologie locale à deux sauts afin de générer un arbre de recouvrement minimal de racine  $x$ , où les MPRs-2 sont les fils de  $x$  et les voisins à deux sauts sont les feuilles. La complexité de cet algorithme est de  $O(\alpha)$ , où  $\alpha$  présente le nombre maximum des voisins à deux sauts d'un nœud dans le graphe.

Nous montrons maintenant que l'utilisation des MPRs-2 dans QOLSR permet de trouver des chemins dans une topologie partielle qui ont les mêmes performances que ceux dans la topologie entière. Autrement dit, la topologie partielle diffusée contient les meilleurs liens entre les nœuds.

**Lemme 6.3.1** Soit  $p = (a_0, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_k, a_{k+1})$  le chemin le plus large de référence entre deux nœuds  $a_0$  et  $a_{k+1}$ ,  $k \geq 1$ . Pour chaque nœud intermédiaire  $a_i$  ( $i \neq 0$  et  $i \neq k+1$ ) dans le chemin  $p$  qui n'est pas un MPR-2 de son prédécesseur  $a_{i-1}$ , il existe un nœud  $b_i$  sélectionné comme MPR-2 par  $a_{i-1}$  tel que le chemin  $(a_0, a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_k, a_{k+1})$  offre la même bande passante.

PREUVE :

Soit  $p = (a_0, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_k, a_{k+1})$  le chemin le plus large de référence entre la source  $a_0$  et la destinataire  $a_{k+1}$ ,  $k \geq 1$  (figure 6.4).

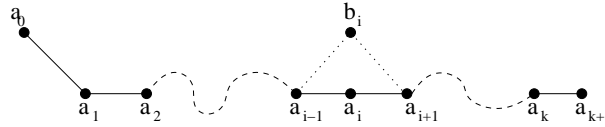


FIG. 6.4 – Le chemin le plus large de référence entre  $a_0$  et  $a_{k+1}$ .

Supposons qu'il existe un nœud  $a_i$  dans le chemin  $p$  qui n'est pas sélectionné comme MPR-2 par son prédécesseur  $a_{i-1}$ . Le nœud  $a_i$  est un voisin à 1-saut de  $a_{i-1}$ .  $a_{i+1}$  est un voisin à 1-saut de  $a_i$  et à 2-sauts de  $a_{i-1}$ . En se basant sur l'idée que tous les voisins à 2-sauts doivent être couverts par au moins un nœud de l'ensemble des MPRs-2 et comme  $a_i$  n'est pas un MPR-2 de  $a_{i-1}$  alors  $a_{i-1}$  doit avoir forcément un autre voisin direct  $b_i$  sélectionné comme MPR-2 pour atteindre  $a_{i+1}$ . Soit  $p' = (a_0, a_1, \dots, a_{i-1}, b_i, a_{i+1}, \dots, a_k, a_{k+1})$ ,  $k \geq 2$ . Selon l'algorithme de sélection des MPRs-2 montré précédemment,  $a_{i-1}$  sélectionne  $b_i$  comme MPR-2 pour atteindre  $a_{i+1}$  au lieu de  $a_i$  pour les deux critères suivants :

$$\text{Bw}(a_{i-1}, b_i, a_{i+1}) > \text{Bw}(a_{i-1}, a_i, a_{i+1}) \quad (6.1)$$

Ou

$$\begin{cases} \text{Bw}(a_{i-1}, b_i, a_{i+1}) = \text{Bw}(a_{i-1}, a_i, a_{i+1}) \\ \text{del}(a_{i-1}, b_i, a_{i+1}) < \text{del}(a_{i-1}, a_i, a_{i+1}) \end{cases} \quad (6.2)$$

À partir de l'équation 6.1, on a

$$\begin{cases} \text{Bw}(p') \geq \text{Bw}(p) \\ \text{Il n'y a aucune garantie que } \text{del}(p') \geq \text{del}(p) \end{cases} \quad (6.3)$$

À partir de l'équation 6.2, on a

$$\begin{cases} \text{Bw}(p') = \text{Bw}(p) \\ \text{del}(p') < \text{del}(p) \end{cases} \quad (6.4)$$

Dans les deux équations 6.3 et 6.4, nous avons  $\text{Bw}(p') \geq \text{Bw}(p)$ . Puisque le chemin  $p$  et le plus large chemin de référence par hypothèse, on obtient  $\text{Bw}(p') = \text{Bw}(p)$  et donc le chemin  $p'$  est aussi un plus large chemin de référence où chaque nœud intermédiaire est un MPR-2 de son prédécesseur.

◆

**Lemme 6.3.2** *Soit  $p$  le chemin le plus large de référence entre une paire de nœuds calculé en utilisant la topologie entière. Ce chemin  $p$  possède la caractéristique que chacun de ses nœuds intermédiaires est un MPR-2 de son prédécesseur, sinon il existe un chemin équivalent en terme de bande passante où chacun de ses nœuds intermédiaires est un MPR-2 de son prédécesseur.*

PREUVE :

Ce lemme peut être montré facilement par récurrence. Soit  $p = (s, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_k, t)$  le chemin le plus large de référence entre  $s$  et  $t$  (figure 6.5).

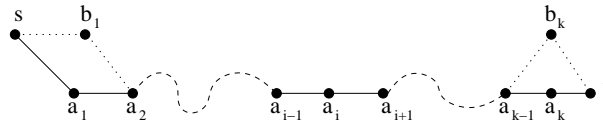


FIG. 6.5 – Le chemin le plus large de référence entre  $s$  et  $t$ .

a) Nous démontrons que le premier nœud intermédiaire  $a_1$  est sélectionné comme MPR-2 par la source  $s$ . Nous distinguons deux cas : i)  $a_1$  est un MPR-2 de  $s$  et donc le nœud MPR-2 de la source est dans le chemin le plus large de référence, ii) sinon en utilisant le lemme 6.3.1, il existe un nœud  $b_1$  sélectionné comme MPR-2 par  $s$  tel que le chemin  $p' = (s, b_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_k, t)$  a la même bande passante que le chemin  $p$  ( $p'$  est aussi un large chemin de référence). Alors, le nœud MPR-2 de la source est dans le chemin le plus large de référence.

b) Nous supposons que tous les nœuds  $\{a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{k-1}\}$  sont sélectionnés chacun comme un MPR-2 par leurs prédécesseurs dans le chemin  $p$ , et nous démontrons que le nœud suivant de  $a_{k-1}$  dans le chemin le plus large de référence vers  $t$  est un MPR-2 de  $a_{k-1}$ . Nous distinguons deux cas : i)  $a_k$  est un MPR-2 de  $a_{k-1}$  et donc le nœud suivant de  $a_{k-1}$  dans le chemin le plus large de référence est un MPR-2, ii) si le nœud  $a_k$  n'est pas un MPR-2 de  $a_{k-1}$ , en utilisant le lemme 6.3.1, il existe un nœud  $b_k$  sélectionné comme MPR-2 par  $a_{k-1}$  tel que le chemin  $p' = (s, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{k-1}, b_k, t)$  a la même bande passante que le chemin  $p$  ( $p'$  est aussi un large chemin de référence). Le cas (i) montre que chaque nœud intermédiaire dans le chemin  $p$  le plus large par référence est sélectionné comme MPR-2 par son prédécesseur dans le même chemin. Sinon, le cas (ii) montre l'existence d'un chemin alternatif où les nœuds intermédiaires sont des MPRs-2 de leurs prédécesseurs dans le même chemin.

◆

En utilisant le protocole QOLSR, chaque nœud dans le réseau construit sa carte topologique partielle avec les liens entre chaque nœud et ses MPRs-2. L'algorithme de routage utilisé dans

QOLSR peut calculer facilement les chemins les plus larges dans cette topologie partielle. Le lemme 6.3.2 montre que le chemin le plus large calculé par la source vers une destination possède la même performance que celui calculé dans la topologie entière. Par déduction, on a le théorème suivant :

**Théorème 6.3.3** *L'utilisation de l'heuristique de sélection des MPRs-2 dans le protocole QOLSR permet de trouver les chemins les plus larges de références.*

L'ensemble des MPRs-2 est recalculé dans les trois cas suivants :

- un changement dans le voisinage direct est détecté : l'apparition ou la disparition d'un lien bidirectionnel ou un lien bidirectionnel devient unidirectionnel.
- un changement dans le voisinage à deux sauts est détecté : l'apparition ou la disparition d'un lien bidirectionnel.
- un changement dans la bande passante ou le délai sur un lien direct ou à deux sauts est détecté. En effet, chaque nœud mesure le pourcentage de changement de la bande passante et le délai entre l'ancienne et la nouvelle valeur calculé sur un intervalle de temps. Si ce pourcentage dépasse un seuil BANDWIDTH\_THRESHOLD pour la bande passante ou DELAY\_THRESHOLD pour le délai, le nœud doit recalculer ses MPRs-2.

### 6.3.6 La déclaration de la topologie et les conditions de QoS

Dans cette section, nous décrivons le contenu des informations de contrôle et comment elles sont utilisées pour construire les routes. La détection de voisinage avec les valeurs de QoS sur les liens permet à chaque nœud de communiquer avec ses voisins directs et choisir ses MPRs-2 qui lui permettent de disséminer ses informations de contrôle dans le réseau. Les routes sont construites en utilisant les MPRs-2 et les liens directs vers les voisins à 1 et 2-sauts.

#### Base d'informations de la topologie et les conditions de QoS

Chaque nœud dans le réseau maintient une base d'informations de la topologie du réseau. Ces informations sont collectées après l'analyse des messages *TC* reçus par ce nœud.

Pour chaque destination dans le réseau, le nœud maintient le tuple suivant [T\_dest, T\_last, T\_bandwidth, T\_delay, T\_met\_1, ..., T\_met\_n, T\_seq, T\_time] où T\_dest est l'adresse de la destination qui est à 1 saut du nœud avec l'adresse T\_last. Autrement dit, T\_last est un MPR-2 de T\_dest. T\_bandwidth, T\_delay, T\_met\_1, ... et T\_met\_n désignent la bande passante disponible, le délai et les n-additionnelles métriques respectivement sur le lien entre T\_dest et T\_last. T\_seq est le numéro de séquence, et T\_time est le temps au bout duquel ce tuple expire et doit être détruit.

#### Le nouveau format d'un message *TC*

Une extension est ajoutée aux messages *TC* pour supporter la QoS dans MANETs. Tous les nœuds choisis comme MPRs-2 doivent diffuser périodiquement dans le réseau des messages *TC*.

Le nouveau format du message *TC* est explicité en annexe A.2.

### Le traitement d'un message TC

Les messages *TC* sont diffusés dans le réseau entier en utilisant la diffusion optimisée de QOLSR par l'intermédiaire des MPRs-2. À la réception d'un message *TC* non encore traité, chaque nœud enregistre les informations contenues dans ce paquet dans sa table topologique selon le processus décrit dans [73] en rajoutant pour chaque entrée les informations de QoS sur le lien ( $T\_dest, T\_last$ ).

#### 6.3.7 Calcul des routes

Chaque nœud dans le réseau calcule pour chaque flux QoS le meilleur chemin selon la demande de QoS exigée. Le calcul de la table de routage est basé sur les informations contenues dans la base d'informations de voisinage ainsi que celui de la base de topologie. De ce fait, chaque fois que l'une de ses bases d'informations change, les routes doivent être recalculées. Une entrée dans la table de routage possède le format suivant :  $[R\_dest, R\_next, R\_dist, R\_bandwidth, R\_delay, R\_met\_1, \dots, R\_met\_n]$  où  $R\_dest$  est l'adresse du nœud destinataire à atteindre après  $R\_dist$  sauts en passant d'abord par le voisin direct  $R\_next$ .  $R\_bandwidth, R\_delay, R\_met\_1, \dots$  et  $R\_met\_n$  représentent les valeurs des paramètres de QoS sur le chemin vers la destination finale  $R\_dest$ .

Pour un flux best-effort, la route est calculée en utilisant l'algorithme des chemins shortest-widest (voir chapitre 3). Le résultat de cet algorithme représente le maximum qu'un nœud peut offrir afin de mieux router les flux best-effort. Le chemin shortest-widest est calculé en se basant sur la bande passante restante non réservée par les nœuds intermédiaires. Une table de routage est déjà calculée avant l'arrivée des flux best-effort. Cette table de routage est mise à jour à chaque changement topologique ou conditions de QoS dans le réseau.

Concernant les flux QoS, les routes sont calculées immédiatement et localement par la source à la demande en utilisant le graphe topologique pondéré et l'algorithme de la relaxation lagrangienne. Nous avons vu dans le chapitre précédent que le problème de trouver une route sous plusieurs métriques est un problème NP-complet et on a proposé un algorithme RLMA qui peut résoudre des problèmes avec deux, trois ou quatre métriques. Si le flux QoS demande un seuil pour la bande passante et un autre pour le délai ou la gigue par exemple, QOLSR transforme par défaut le problème en un problème de minimisation de nombre de sauts ou du coût sous les contraintes exigées. En suite, QOLSR fait appel à l'algorithme RLMA. Le chemin entier est conservé pour procéder à un contrôle d'admission et même lors de routage pour éviter le problème d'oscillation.

#### 6.3.8 Oscillation des flux

Soit le graphe topologique de la figure 6.6. Il existe deux chemins  $p_1 = (s, a, b, c, t)$  et  $p_2 = (s, d, e, f, t)$  entre la source  $s$  et la destination  $t$ . Les bandes passantes libres des deux chemins sont données par :

$$\begin{cases} \Gamma(p_1) = \min\{\Gamma_{sa}, \Gamma_{ab}, \Gamma_{bc}, \Gamma_{ct}\} \\ \Gamma(p_2) = \min\{\Gamma_{sd}, \Gamma_{de}, \Gamma_{ef}, \Gamma_{ft}\} \end{cases}$$

On suppose que  $\Gamma(p_1) > \Gamma(p_2)$ . Si le nœud  $s$  veut transmettre un flux  $f_1$  avec une demande de bande passante  $\Delta_s$  tel que,  $\Gamma(p_1) - \Delta_s < \Gamma(p_2)$  (il est clair que  $\Gamma(p_2) - \Delta_s < \Gamma(p_2)$ ), il choisit le chemin  $p_1$  puisque ce dernier offre plus de bande passante que le chemin  $p_2$ . Le nœud  $s$  crée une entrée dans sa table de routage pour le flux  $f_1$  avec  $t$  comme destination finale et  $a$  comme le prochain saut. Pendant la transmission de  $f_1$ , la nouvelle bande passante libre du chemin  $p_1$  devient

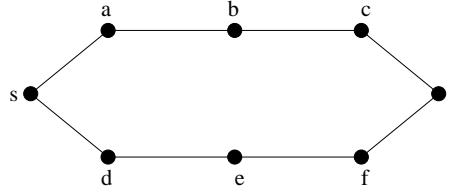


FIG. 6.6 – *L'oscillation est possible sur ce graphe topologique.*

$\Gamma(p_1) - \Delta_s$ . Puisque  $\Gamma(p_1) - \Delta_s < \Gamma(p_2)$ , le nœud  $s$  et selon le protocole QOLSR doit recalculer une route maximale en termes de bande passante. Alors,  $s$  choisi le chemin  $p_2$ . Le prochain saut de  $s$  pour le flux  $f_1$  devient  $d$ . En envoyant les flux sur le chemin  $p_2$ , la bande passante de ce chemin devient  $\Gamma(p_2) - \Delta_s$  ce qui oblige la source à changer de nouveau son prochain saut de  $d$  à  $a$  et ainsi de suite. Le chemin du flux  $f_1$  est recalculé inutilement sans arrêt. Ce problème présente l'*oscillation des flux*.

Les messages *TC* apportent à un nœud source des informations sur les changements des conditions de QoS dans le réseau. Malheureusement, cette source ne peut pas savoir si ce changement est dû à sa transmission puisqu'elle ne garde pas en mémoire le chemin complet d'un flux et rien ne permet de garantir alors qu'un flux est bien passé par le chemin calculé, au-delà de nœud suivant.

Afin d'éviter le problème d'oscillation, les liens d'une route trouvée pour un flux QoS sont maintenus pour détecter facilement l'état de cette route en termes de connexion et de QoS à partir des informations contenues dans les messages *TC*. Cette solution facilite aussi la détection des problèmes sur une route réservée pour un flux QoS par la suite. Pour le nœud  $s$  de l'exemple précédant, il suffit qu'il rajoute  $\Delta_s$  sur les liens de son chemin pour ne pas changer son prochain saut.

## 6.4 Estimation des métriques

La bande passante et le délai d'un lien sont des métriques obligatoires dans le fonctionnement de QOLSR. Ces métriques sont calculées en permanence et diffusées dans le réseau afin de répondre à la nature proactive de QOLSR. Un graphe topologique pondéré doit être disponible au moment de calcul d'une route pour un flux. Pour mesurer ces métriques, les messages de contrôles de QOLSR et le trafic de données sont utilisés. Nous considérons que chaque mobile est équipé par une carte sans fil 802.11 et alors nous utilisons la norme IEEE 802.11 comme la couche MAC. Nous avons proposé plusieurs méthodes analytiques et expérimentales.

### 6.4.1 Une méthode analytique pour le calcul du délai

Le protocole QOLSR est proactif, donc le délai sur un lien doit être fourni même avant l'émission du trafic de données. La seule solution possible est d'utiliser les messages de contrôle de QOLSR qui circulent en permanence dans le réseau. Les messages de contrôle *Hello*s sont périodiques et beaucoup plus fréquents que les messages topologiques *TC*. Par conséquent, nous allons utiliser ces messages pour estimer le délai sur un lien qui sera présenté plus tard lors d'un calcul d'une route comme le délai d'un paquet de données sur ce même lien. Un message *Hello* est envoyé en mode diffusion sans ACKs alors la transmission d'un paquet de données est en mode unicast avec réception d'ACK. La technique consiste à représenter le temps d'émission d'un paquet de données par le temps d'émission d'un message *Hello* et le temps de réception d'un ACK

de voisin par le temps de réception d'un message *Hello* émis par ce voisin.

Chaque nœud utilise les messages *Hello* envoyés et reçus pour calculer le délai total moyen de transmission (ETTT : *Expected Total Time Transmission*). ETTT dépend de plusieurs paramètres : le temps de séjour dans la couche MAC, la perte des paquets due aux interférences et aux congestions ainsi le débit de la carte sans fil. Le temps de séjour dans le buffer de la couche MAC est un élément critique à maintenir sous un certain seuil afin d'éviter le point de saturation qui génère des pertes des paquets et des chutes de performances. Nous allons voir par la suite que les mécanismes de contrôle d'admission et congestion sont importants pour éviter les points de saturation.

Le temps de service résume la transmission effective d'un paquet (support est libre) qui peut prendre un temps non négligeable en présence des interférences dues aux transmissions dans le voisinage (pas forcément les voisins à 1-saut ou 2-sauts). Dans ce cas, le protocole CSMA/CA procède à des retransmissions qui peuvent échouer si le problème persiste et un rejet de paquet est inévitable au bout d'un certain nombre de retransmissions.

À partir du *driver* de la carte sans fil, le nombre des paquets dans le buffer peut être extrait. Le temps moyen de séjour dans le buffer sera exprimé en fonction du nombre moyen des paquets ce buffer (ANP : *Average Number of Packets in queue*). Donc, ETTT est donné par la formule suivante :

$$ETTT = ANP * EST, \quad (6.5)$$

où EST est le temps moyen de service (*Expected Service Time*) basé sur l'algorithme de backoff de IEEE 802.11.

Une fois le temps total moyen de transmission ETTT ( $ETTT = ANP * EST$ ) est calculé et afin d'éviter la variation entre deux mesures, chaque nœud doit étaler et corriger le délai mesuré en l'intégrant avec les anciennes mesures. Comme dans le mécanisme de contrôle de congestion dans TCP, le temps total moyen de transmission corrigé (SETTT : *Smoothed ETTT*) est calculé de la manière suivante :

$$SETTT = \alpha \times SETTT + (1 - \alpha) \times ETTT \quad (6.6)$$

où  $\alpha$  est un facteur de correction. Dans les simulations [5], il est montré que les meilleures performances sont réalisées par  $\alpha = 0.4$ .

Maintenant, nous montrons la méthode analytique pour calculer EST. Cette méthode nécessite la probabilité de perte  $p$  sur un lien ainsi que les probabilités d'état du canal sans fil.

### Calcul de la probabilité de perte sur un lien

Soit  $l$  un lien reliant un nœud  $x$  et son voisin  $y$ . Soit ETC (*Expected Transmission Count*) le nombre moyen nécessaire des tentatives de transmission sur un lien pour avoir enfin une transmission réussite. Il est clair que les retransmissions sont aussi incluses dans le ETC. Le ETC d'une route est la somme de tous les ETCs des liens. Par exemple, le ETC d'une route parfaite à trois sauts est de 3 ; le ETC d'une route à un saut avec 50% comme taux de délivrance est égale à 2. Le ETC d'un lien est calculé en utilisant les taux de délivrance aller et retour sur un lien. Le taux de délivrance sur un lien est calculé à partir de la probabilité de perte des paquets dans les deux sens (aller et retour). Soit  $p_f$  la probabilité de perte d'aller et  $p_r$  celle de retour. Une transmission est considérée comme réussite si l'émetteur reçoit un ACK pour cette transmission. Soit  $p$  la probabilité d'une transmission non réussite (collision) de  $x$  à  $y$ . La probabilité de perte  $p$  est donnée par

$$p = 1 - (1 - p_f) \cdot (1 - p_r). \quad (6.7)$$



La couche MAC doit retransmettre le paquet en cas de collision. Soit  $s(k)$  la probabilité d'avoir une transmission réussite après  $k$  tentatives de transmission. On a alors

$$s(k) = p^{k-1} \cdot (1 - p). \quad (6.8)$$

Puisque chaque tentative de transmission est considérée comme un tirage de Bernoulli, le nombre moyen de transmission est donnée par

$$\text{ETC} = \frac{1}{1 - p}. \quad (6.9)$$

Plus formellement,

$$\text{ETC} = \sum_{k=1}^{+\infty} k \cdot s(k) = \sum_{k=1}^{+\infty} k \cdot p^{k-1} \cdot (1 - p) = (1 - p) \sum_{k=1}^{+\infty} k \cdot p^{k-1} = (1 - p) \frac{1}{(1 - p)^2} = \frac{1}{1 - p}. \quad (6.10)$$

On remarque que l'équation 6.9 suppose que la probabilité de collision est indépendante de la taille d'un paquet et identiquement distribuée. Ceci peut être vrai dans le cas où le schéma four-way handshake est utilisé. En effet, le mécanisme RTS/CTS permet de réduire significativement le problème des nœuds cachés puisque ces nœuds sont au courant via la réception d'un message CTS qu'un nœud a réservé le support durant la transmission. Donc, si une transmission est réussie sur le premier slot, elle aura plus de chance de réussir sur le reste des slots indépendamment de la taille du paquet. Cette vision ne s'applique pas dans le cas d'utilisation d'un schéma two-way handshake où chaque nœud est vulnérable aux transmissions des nœuds cachés ce qui implique une forte dépendance à la taille de paquet.

Les messages de contrôle sont aussi vulnérables au problème des nœuds cachés puisque le broadcast s'effectue sans réservation du support et sans retransmission en cas de collision.

L'échange des messages *Hello* permet de calculer les probabilités de pertes  $p_f$  et  $p_r$ . Chaque nœud envoie périodiquement des messages *Hello* à chaque intervalle de temps *hello\_interval* (2 seconds). Afin d'éviter le problème de synchronisation des transmissions, chaque nœud doit soustraire un temps *jitter* de *hello\_interval*. Ce *jitter* est généré aléatoirement pour chaque message *Hello* généré. L'intervalle de génération devient :

$$\text{Actual hello interval} = \text{hello\_interval} - \text{jitter}$$

où la valeur de *jitter* est sélectionnée aléatoirement et uniformément dans un intervalle  $[0, \text{MAXJITTER}]$ . L'implémentation d'OLSR propose une valeur  $\text{MAXJITTER} = \text{hello\_interval}/4$ . Une approximation de valeur de jitter est donnée par

$$\text{jitter} = \frac{\text{MAXJITTER}}{2}.$$

Puisque les messages *Hello* sont en mode broadcast localement, alors ils ne sont pas acquittés ni retransmis. Chaque nœud doit donc compter pour chacun de ses voisins le nombre des messages *Hello* reçus dans une fenêtre de  $w$  secondes. À chaque instant  $t$ , la probabilité de perte ( $L_p$  : *Loss probability*) est donnée par

$$L_p(t) = \frac{\text{count}(t - w, t)}{w / (\text{hello\_interval} - \text{MAXJITTER}/2)}, \quad (6.11)$$

où  $\text{count}(t - w, t)$  est le nombre des messages *Hello* reçus dans un intervalle de temps  $w$ , et  $w$  divisé par  $(\text{hello\_interval} - \text{MAXJITTER}/2)$  présente le nombre total des messages *Hello* qui doivent

être reçus dans la fenêtre  $w$ . Pour le lien  $x \rightarrow y$ , le nœud  $y$  calcule la probabilité  $p_f$ . Le nœud  $x$  de son tour, calcule la probabilité  $p_r$  de  $y$  sur le lien  $y \rightarrow x$ . Ensuite, la probabilité  $p_f$  est intégrée dans le message *Hello* envoyé par  $y$  et  $p_r$  dans le message *Hello* envoyé par  $x$ . À la réception de ces messages, chacun des nœuds  $x$  et  $y$  auront les deux probabilités pour enfin calculer la probabilité de perte  $p$ .

L'utilisation de la probabilité de perte des messages *Hello* ne représente pas réellement la probabilité de perte des données : i) les messages *Hello* ne sont pas retransmis en cas de collision. Par conséquent, les retransmissions ne sont pas exprimées dans la probabilité de perte ; ii) si le mécanisme RTS/CTS est utilisé, la probabilité de perte des données est moins vulnérable au problème des nœuds cachés tandis que la probabilité de perte des messages *Hello* dépend de ce problème ; iii) si le mécanisme de base est utilisé, la probabilité de perte des données dépend de la taille des paquets de données en raison de présence des nœuds cachés tandis que la probabilité de perte des messages *Hello* dépend uniquement de la taille de ses messages.

### Calcul de temps moyen de service (EST)

Pour calculer le temps moyen de service d'un paquet de données (EST : *Expected Service Time*), une modélisation du protocole IEEE 802.11 est nécessaire. Cette modélisation permet d'exprimer d'une manière simple et exacte le fonctionnement de l'algorithme de Backoff et la probabilité de réaliser une transmission réussite après  $k$  tentatives.

L'algorithme de Backoff définit une fenêtre de contention (CW : Contention Window) qui correspond au nombre de time-slots qui peuvent être sélectionnés pour le calcul de temps d'attente (appelé temporisateur de Backoff). La figure 6.7 montre un exemple de fonctionnement de l'algorithme de Backoff.

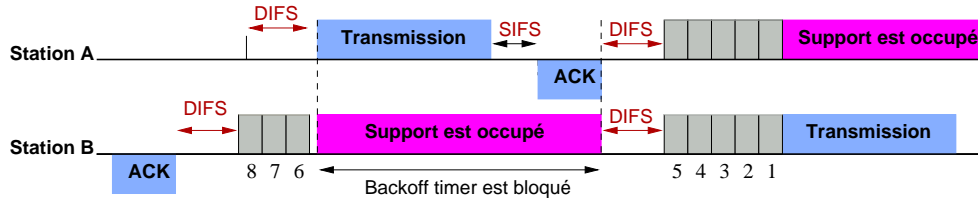


FIG. 6.7 – L'algorithme de Backoff.

Lorsque le support est libre, le nœud décrémente son timer jusqu'à ce que le support soit occupé ou jusqu'à ce que le timer atteigne la valeur 0. Si le support devient libre pendant une période supérieure à un DIFS après le blocage de la décrémentation (i.e., le support a été occupé), le nœud peut à nouveau faire décrémente son Backoff timer. Lorsque le Backoff timer est bloqué, seuls deux événements exclusifs et mutuels peuvent se produire dans le canal : soit une transmission réussite d'un nœud dans le voisinage occupe le canal, soit une collision. En effet, le temps qui sépare deux décrémentations successives (étape de Backoff) peut passer uniquement par un de ces trois événements exclusifs  $E_s = \{\text{transmission réussite}\}$ ,  $E_i = \{\text{canal libre}\}$  and  $E_c = \{\text{collision}\}$ . En d'autre terme, durant le Backoff d'un nœud, la  $j$ -ème étape de Backoff se produit sur une collision, transmission réussite ou le canal écouté est libre. Une transmission réussite ou collision sont suivies toujours par un temps DIFS ce qui implique une décrémentation de Backoff timer et c'est pour cette raison qu'un et un seul événement est possible dans un slot Backoff. Les événements successifs durant le Backoff sont indépendants. Dans le mode DCF, un nœud décide que sa transmission a subi une collision s'il ne reçoit pas un ACK au-delà d'un certain timeout : un ACK

timeout pour le mécanisme de base et CTS timeout dans le mécanisme RTS/CTS. Autrement dit, si une collision se produit dans une étape de Backoff, nous supposons que les nœuds en collision se lancent dans le processus de résolution dans la même *étape de Backoff* et donc ils peuvent être prêts pour une transmission dès la prochaine *étape de Backoff*. De cette façon, nous évitons le problème de dépendance au nombre des nœuds en collision de l'*étape de Backoff* précédente.

Nous supposons que les événements successifs sont indépendants et mutuellement exclusifs dans chaque *étape de Backoff*. Soit  $k$  le  $k$ -ème étage de Backoff ( $k$ -ème tentative de transmission) et  $n_k$  le nombre des slots de Backoff timer choisit aléatoirement lors de la  $k$ -ème étage de Backoff. Soient  $p_s = P\{E_s\}$ ,  $p_i = P\{E_i\}$  et  $p_c = P\{E_c\}$  les probabilités d'avoir respectivement une transmission réussie, canal libre et collision.  $n_k$  peut avoir  $r_i$  slots libre,  $r_c$  slots de collision et  $r_s$  slots de transmission réussie tels que  $r_i + r_c + r_s = n_k$  et  $p_i + p_c + p_s = 1$ . Soient  $\sigma$  l'unité de temps utilisé lorsque le canal est libre (i.e., un slot de Backoff timer),  $t_s$  le temps moyen pendant lequel le canal est occupé par une transmission réussie et  $t_c$  le temps moyen pendant lequel le canal est occupé par une collision. Le temps total de Backoff au  $k$ -ème étage de Backoff noté  $CW(n_k)$  est donné par

$$CW(n_k) = \sigma r_i + t_c r_c + t_s r_s. \quad (6.12)$$

Il est clair que le Backoff timer atteint la valeur 0 après  $\sigma r_i + t_c r_c + t_s r_s$  temps de slot.

Le temps moyen du  $k$ -ème étage de Backoff lorsque  $n_k$  slots sont choisis est simplement

$$\overline{CW(n_k)} = n_k(\sigma p_i + t_c p_c + t_s p_s). \quad (6.13)$$

Le nombre de slots  $n_k$  de la  $k$ -ème étage est tiré uniformément dans un intervalle  $[0, W_k - 1]$ . Le nombre de slot du  $k+1$ -ème étage est sélectionné dans l'intervalle  $[0, (2 * W_k) - 1]$ . Puisque le tirage se fait selon la loi uniforme, le nombre moyen des slots de la  $k$ -ème étage est de  $(W_k - 1)/2$ . On peut calculer le temps moyen de Backoff  $\overline{CW(k)}$  du  $k$ -ème étage en se basant sur la moyenne de  $n_k$  :

$$\overline{CW(k)} = \sum_{n_k=0}^{W_k-1} \overline{CW(n_k)} P\{n_k\} = \sum_{n_k=0}^{W_k-1} n_k(\sigma p_i + t_c p_c + t_s p_s) / W_k = \vartheta (W_k - 1) / 2, \quad (6.14)$$

où  $\vartheta = \sigma \cdot p_i + t_c \cdot p_c + t_s \cdot p_s$ . Ce dernier résultat est presque intuitif : le temps moyen de Backoff du  $k$ -ème étage de Backoff n'est que le produit du nombre moyen des *étapes de Backoff*,  $(W_k - 1)/2$ , et le temps moyen d'une *étape de Backoff*,  $\vartheta$ .

Maintenant, nous considérons le cas le plus général de l'algorithme exponentiel du Backoff. Le temps total de Backoff (TBT : *Total Backoff Time*) est calculé de la manière suivante :

$$TBT(n_k) = \sum_{i=1}^{k-1} (CW(n_i) + t_c) + CW(n_k) = \sum_{i=1}^k CW(n_i) + (k-1)t_c, \quad (6.15)$$

où  $t_c$  est le temps moyen de découvrir une collision et de passer à l'étape de résolution.  $CW(n_i)$  présente le temps de Backoff lors d' $i$ -ème étage de Backoff. En utilisant le temps moyen de Backoff  $\overline{CW(n_i)}$  sur  $n_i$  slots d' $i$ -ème étage, le temps moyen de Bckoff  $\overline{TBT}(n_k)$  calculé sur la base de temps moyen de Backoff est donné par

$$\overline{TBT}(n_k) = \sum_{i=1}^k \overline{CW(n_i)} + (k-1)t_c. \quad (6.16)$$

En utilisant la moyenne sur les  $n_k$  slots, on obtient

$$\overline{TBT}(k) = \sum_{i=1}^k \overline{CW}(i) + (k-1)t_c. \quad (6.17)$$

Le temps total moyen de Backoff  $\overline{TBT}$  est donné par

$$\begin{aligned} \overline{TBT} &= \sum_{k=1}^{\infty} \overline{TBT}(k) P\{K=k\} = \sum_{k=1}^{\infty} \left[ \sum_{i=1}^k \overline{CW}(i) + (k-1)t_c \right] s(k) \\ &= \sum_{k=1}^{\infty} \left[ \sum_{i=1}^k \left( \frac{\vartheta}{2} W_i \right) p^{k-1} (1-p) \right] - \frac{\vartheta}{2(1-p)} + \frac{p}{1-p} t_c \\ &= \Phi - \frac{\vartheta}{2(1-p)} + \frac{p}{1-p} t_c, \end{aligned} \quad (6.18)$$

où

$$\Phi = \sum_{k=1}^{\infty} \left[ \sum_{i=1}^k \left( \frac{\vartheta}{2} W_i \right) p^{k-1} (1-p) \right].$$

Pour calculer le premier terme ( $\Phi$ ) dans l'équation 6.18, nous observons d'abord que

$$W_i = \begin{cases} 2^{i-1} W_{\min} & \text{if } 1 \leq i \leq m \\ 2^m W_{\min} & \text{if } i > m, \end{cases} \quad (6.19)$$

où  $m$  est le maximum des étages de Backoff ou bien la valeur pour laquelle  $W_{\max} = 2^m W_{\min}$  (les valeurs de  $W_{\min}$  et  $W_{\max}$  sont fixées par la couche physique). Le reste de la somme dans  $\Phi$  peut être divisé en deux termes :

$$\begin{aligned} \Phi &= \sum_{k=1}^{\infty} \left( \sum_{i=1}^k \left( \frac{\vartheta}{2} W_i \right) \right) p^{k-1} (1-p) \\ &= \sum_{k=1}^m \left( \sum_{i=1}^k \left( \frac{\vartheta}{2} W_i \right) \right) p^{k-1} (1-p) + \sum_{k=m+1}^{\infty} \left( \sum_{i=1}^k \left( \frac{\vartheta}{2} W_i \right) \right) p^{k-1} (1-p) \\ &= S_1 + S_2. \end{aligned}$$

Pour  $S_1$ , on a

$$S_1 = \sum_{k=1}^m \left( \sum_{i=1}^k \left( \frac{\vartheta}{2} W_i \right) \right) p^{k-1} (1-p) = \frac{\vartheta}{2} W_{\min} \left\{ 2(1-p) \left[ \frac{1-(2p)^m}{1-2p} \right] + p^m - 1 \right\}.$$

Pour trouver  $S_2$ , nous notons d'abord que pour  $k = m+1$ ,

$$\sum_{i=1}^k \frac{\vartheta}{2} W_i = \frac{\vartheta}{2} W_{\min} \sum_{i=1}^{m+1} 2^{i-1} = \frac{\vartheta}{2} W_{\min} (2^{m+1} - 1).$$

Pour  $k = m+2$ ,

$$\sum_{i=1}^k \frac{\vartheta}{2} W_i = \frac{\vartheta}{2} W_{\min} \left[ \sum_{i=1}^{m+1} 2^{i-1} + \sum_{i=m+2}^{m+2} 2^m \right] = \frac{\vartheta}{2} W_{\min} [(2^{m+1} - 1) + 2^m].$$

En général, pour  $k = m + \delta$ ,

$$\sum_{i=1}^k \frac{\vartheta}{2} W_i = \frac{\vartheta}{2} W_{\min} \left[ \sum_{i=1}^{m+1} 2^{i-1} + \sum_{i=m+2}^{m+\delta} 2^m \right] = \frac{\vartheta}{2} W_{\min} [(2^{m+1} - 1) + (\delta - 1)2^m].$$

Par un changement de variable  $j = k - (m + 1)$  dans  $S_2$ , on obtient

$$S_2 = \frac{\vartheta}{2} W_{\min} \sum_{j=0}^{\infty} [(2^{m+1} - 1) + j2^m] p^{j+m} (1-p) = \frac{\vartheta}{2} W_{\min} \left[ (2^{m+1} - 1)p^m + \frac{2^m p^{m+1}}{1-p} \right].$$

En remplaçant la valeur de  $\Phi$  dans l'équation 6.18, on aura

$$\overline{TBT} = \frac{\vartheta(W_{\min}\beta - 1)}{2(1-p)} + \frac{p}{1-p} t_c, \quad (6.20)$$

où

$$\beta = \frac{(1-p) - 2^m p^{m+1}}{1-2p}. \quad (6.21)$$

Maintenant, on peut calculer facilement le temps moyen de service EST

$$\boxed{\text{EST} = \overline{TBT} + t_s}, \quad (6.22)$$

où  $t_s$  est le temps moyen d'une transmission réussite. La valeur d'EST dépend des probabilités d'état du canal :  $p_i$ ,  $p_c$  et  $p_s$ , qui ne sont pas faciles à calculer.

### Calcul des probabilités d'état du canal

Nous montrons une piste pour obtenir ces probabilités en se basant sur les travaux de Bianchi [107]. Bianchi propose un modèle théorique qui n'est pas réaliste. Nous l'utilisons pour montrer la façon d'extraire les probabilités d'état de canal de son modèle. Le modèle de Bianchi suppose un nombre fixe  $n$  de nœuds où chaque nœud possède toujours un paquet disponible à transmettre dans son buffer, i.e., le buffer n'est jamais vide. Cette hypothèse est appelée *saturation throughput*. Aussi, il suppose que le canal est idéal sans présence des nœuds cachés. Bianchi modélise le processus stochastique de fonctionnement du Backoff timer par un processus de Markov à temps discret en deux dimensions. Selon son développement, la probabilité  $\gamma$  qu'un nœud transmet dans un slot sélectionné aléatoirement est

$$\gamma = \frac{2(1-2p)}{(1-2p)(W_{\min} + 1) + pW_{\min}(1-(2p)^m)}, \quad (6.23)$$

qui est une fonction de la probabilité de perte présenté au début de cette section. La connaissance de  $p$  permet facilement le calcul de la probabilité  $\gamma$ .

Soit  $P_{tr}$  la probabilité qu'il y'a au moins une transmission dans un slot donné. Puisqu'on considère les événements produits lors du temps Backoff d'un nœud, alors seulement les  $(n-1)$  nœuds restants peuvent entrer en collision lors de l'accès au support. Chacun des  $(n-1)$  nœuds transmet un paquet avec une probabilité  $\gamma$ . On obtient

$$P_{tr} = 1 - (1-\gamma)^{n-1}. \quad (6.24)$$

$p_{tr}$  présente une probabilité forte puisqu'elle considère que tous les nœuds s'entendent. Nous pouvons aussi considérer uniquement les nœuds dans le voisinage qui ont un support commun entre eux. La probabilité d'apparition d'une transmission réussite dans le canal  $P_{suc}$  est donnée par la probabilité d'avoir une seule transmission sur le canal conditionnée l'existence d'au moins une transmission, i.e.,

$$P_{suc} = \frac{\binom{n-1}{1} \gamma (1-\gamma)^{n-2}}{P_{tr}} = \frac{(n-1)\gamma(1-\gamma)^{n-2}}{1 - (1-\gamma)^{n-1}}. \quad (6.25)$$

Maintenant les probabilités d'état du canal sont données par :  $p_s = P\{E_s\} = P_{tr} \cdot P_{suc}$ ,  $p_i = P\{E_i\} = 1 - P_{tr}$  et  $p_c = P\{E_c\} = P_{tr}(1 - P_{suc})$ .

Nous avons montré des méthodes pour calculer la probabilité de perte  $p$ , les probabilités d'état de canal mais il reste les temps  $t_s$  et  $t_c$  afin de calculer la valeur de ETTT.

### Calcul des temps de transmission et collision

Soit  $H = \text{PHY}_{\text{hdr}} + \text{MAC}_{\text{hdr}}$  l'entête du paquet,  $\theta$  le délai de propagation et  $E\{P\}$  la taille moyenne des paquets. Dans un système de base (two-way handshake), les collisions peuvent avoir lieu lors de la transmission des données ou lors de la réception d'ACK.

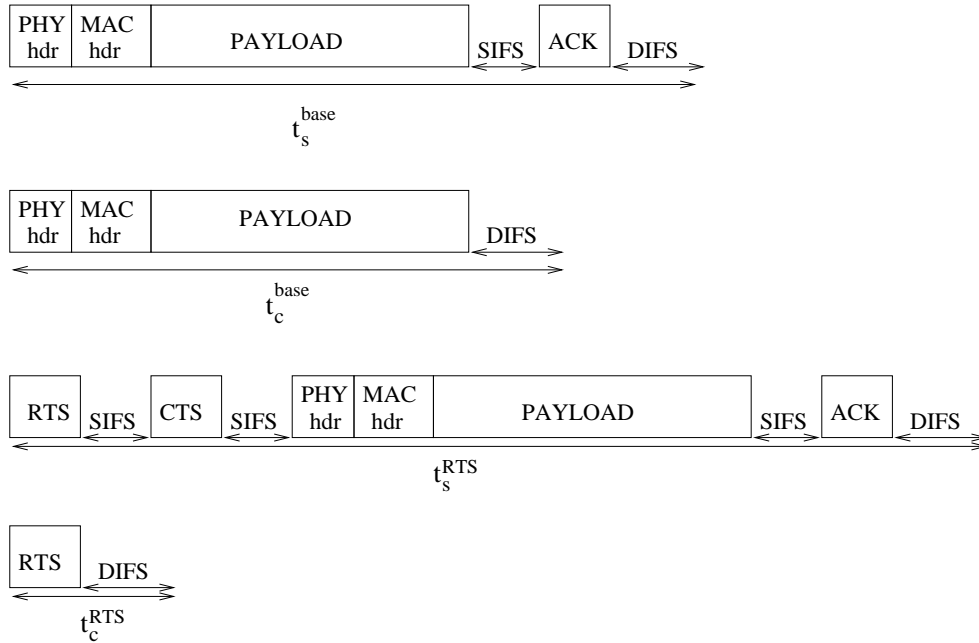


FIG. 6.8 – Les temps  $t_s$  et  $t_c$ .

Selon la figure 6.8

$$t_s^{\text{base}} = H + E\{P\} + \text{SIFS} + \theta + \text{ACK} + \text{DIFS} + \theta. \quad (6.26)$$

$$t_c^{\text{base}} = H + E\{P^*\} + \text{DIFS} + \theta, \quad (6.27)$$

où  $E\{P^*\}$  est la taille moyenne du plus long paquet ayant subi une collision. Dans le cas où tous les paquets ont la même taille,  $E\{P\} = E\{P^*\} = P$ . Le  $t_c$  montré ici est le temps moyen qu'une

station durant son backoff observe une collision sur le support due aux transmissions de deux ou plusieurs stations. La station qui envoie un paquet et qui rentre en collision doit attendre l'ACK timeout avant de découvrir cette collision. Donc, le  $t_c$  des nœuds observants des collisions est petit par rapport au  $t_c$  des stations qui s'impliquent dans la collision.

Dans le cas d'un système complètement géré par le mécanisme d'accès RTC/CTS, seul RTS peut subir la majorité des collisions. Selon la figure 6.8

$$t_s^{RTS} = RTS + SIFS + \theta + CTS + SIFS + \theta + H + E\{P\} + SIFS + \theta + ACK + DIFS + \theta, \quad (6.28)$$

$$t_c^{RTS} = RTS + DIFS + \theta. \quad (6.29)$$

Le  $t_c$  des nœuds observants des collisions est petit par rapport au  $t_c$  des stations qui s'impliquent dans la collision et qui doivent attendre un CTS timeout.

Cette méthode permet de mieux voir le comportement du délai en utilisant le protocole IEEE 802.11 dans un environnement ad hoc mais il est impossible de l'implémenter dans un cas réel.

#### 6.4.2 Une méthode analytique pour le calcul de la bande passante

Dans MANETs, les flux qui traversent la même zone géographique de voisinage sont en compétition pour avoir la meilleure capacité dans le canal sans fil. Des problèmes de collisions, d'interférences et des pertes ralentissent significativement leurs passages. Contrairement aux réseaux filaires où les liens sont indépendants, seulement les flux qui traversent le même lien sont en concurrence. Déterminer la capacité d'un réseau ad hoc est un problème difficile puisque ce problème dépend de plusieurs paramètres : les corrélations entre les liens de voisinage (la corrélation est exprimée en terme d'interférence qui génère des collisions, des retransmissions et même des pertes), le modèle de trafic, la densité des nœuds, la taille du buffer, la mobilité, etc.

Plusieurs travaux ont été proposés pour modéliser la capacité dans MANETs. Dans [108], Gupta et Kumar ont étudié comment la capacité d'un réseau ad hoc statique se comporte asymptotiquement en fonction du nombre des nœuds dans le réseau. Le problème est abordé sous deux modèles différents d'interférence : niveau protocolaire et niveau physique. Dans le modèle au niveau protocolaire, le nœud récepteur reçoit un paquet avec succès envoyé par un nœud émetteur si cet émetteur est géographiquement suffisamment proche que n'importe quel autre nœud émetteur. Avec un modèle au niveau physique, les transmissions des autres nœuds sont considérées comme un bruit, et on suppose qu'une transmission entre un pair source-destination est réussie si le rapport signal sur bruit (SIR : *Signal to Interference Ratio*) est supérieur à un certain seuil. Les auteurs ont montré que sous les deux modèles, la capacité maximale est  $\Theta(1/\sqrt{n})$  par nœud mais avec des positions optimales des nœuds dans le disque unité, i.e., la capacité décroît en  $1/\sqrt{n}$  bits-mètre/s par pair source-destination. Ils ont montré aussi que sous un positionnement aléatoire des nœuds et avec l'utilisation d'un modèle au niveau protocolaire, la capacité est de  $\Theta(1/\sqrt{n \log n})$  par nœud. Par la suite, les travaux de Gupta et Kumar sont améliorés en introduisant la mobilité [109] et la corrélation entre la bande passante effective et le délai [110].

Nous avons précisé précédemment que dans notre protocole QOLSR, la bande passante est calculée sur les liens. Chaque nœud mesure la bande passante disponible sur les liens vers ses voisins. Le trafic de données et les messages de contrôle (*Hello* et *TC*) partagent la bande passante disponible et donc ils sont considérés dans le calcul. La bande passante disponible sur un lien dépend du taux d'utilisation de canal qui exprime l'activité recensée dans le canal (transmission, collision, etc). La quantité et la taille des flux qui passent dans le voisinage ainsi la taille du buffer du nœud récepteur sont des facteurs déterminant pour la bande passante disponible. Soit  $i$  un nœud

et  $j$  son voisin direct avec un lien symétrique. La bande passante disponible sur le lien  $i \rightarrow j$  n'est pas forcément la même que celle sur le lien  $i \leftarrow j$ . Ceci est dû au problème d'interférence qui n'exerce pas les effets d'une manière symétrique sur chacun des nœuds. La bande passante disponible (Bw) sur le lien  $i \rightarrow j$  est calculée par le nœud  $j$  puisque les messages de contrôle envoyés par  $i$  ne sont pas acquittés et il n'y a aucune information s'ils sont bien reçus ou non. Bw est donnée par

$$Bw_{(i \rightarrow j)} = (1 - u) \times BpE_{(i,j)}, \quad (6.30)$$

où  $u$  est la facteure d'utilisation du canal.  $(1 - u)$  représente le taux de repos du canal (canal libre). BpE représente la bande passante effective des transmissions sur le lien calculée sur une fenêtre  $w$  exprimée en nombre de paquets. La BpE d'un paquet est

$$BpE_{\text{paquet}} = \frac{S}{t_q + TBT(n_k) + t_s} \quad (6.31)$$

où  $S$  est la taille du paquet,  $t_q$  est le temps de séjour dans le buffer de l'émetteur,  $TBT(n_k)$  représente le temps total de Backoff (voir section 6.4.1) et  $t_s$  est le temps de transmission avec succès.

Si le nœud  $j$  mesure la BpE pour chaque paquet envoyé par  $i$ , il va avoir des variations considérable dans ses mesures de la bande passante disponible. Cette instabilité oblige le nœud  $i$  à sélectionner fréquemment ses MPRs puisque le nœud  $j$  informe  $i$  sur les changements dans la bande passante. Comme il est mentionné dans [5], l'utilisation d'une fenêtre de paquets augmente la stabilité et la robustesse des mesures. Une fenêtre  $w$  de 16 ou 32 paquets est la plus petite fenêtre efficace qui réalise les meilleures performances (voir la figure 6.9).

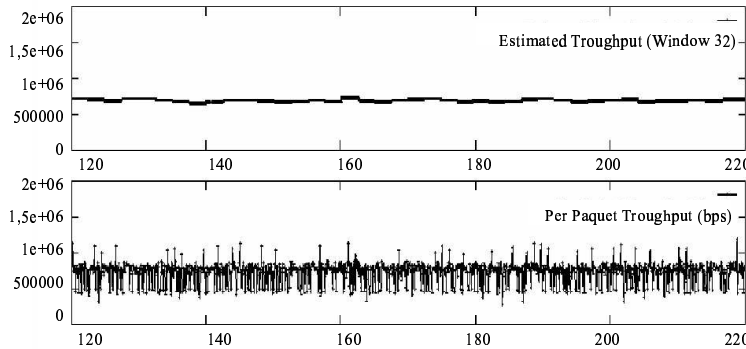


FIG. 6.9 – La BpE par paquet et dans une fenêtre de 32 paquets (source : [5]).

Dans notre cas, cette fenêtre a une double utilité. Elle permet d'assurer la stabilité du réseau d'une part et de calculer le taux de repos du canal d'une autre part. Le nœud  $j$  calcule la moyenne de la taille des 16 ou 32 paquets reçus, ainsi leurs temps de transmission. Ensuite, le rapport taille/temps donne la BpE de la fenêtre.

Chaque nœud peut enregistrer le temps de repos du canal (*idle\_time*) durant la fenêtre  $w$ . Le taux de repos du canal est donné par

$$1 - u = \frac{\text{idle\_time}(w)}{w}. \quad (6.32)$$

Finalement la bande passante disponible sur un lien est obtenue par

$$\frac{\text{idle\_time}(w)}{w} \times \frac{E\{P\}}{SETTT}, \quad (6.33)$$



où  $E\{P\}$  est la taille moyenne des paquets reçus dans la fenêtre  $w$  et SETTT est le temps total moyen de transmission (voir section 6.4.1).

Cette méthode est très difficile à implémenter réellement puisque plusieurs paramètres sont inconnus comme le `idle_time` ou bien le temps de séjour dans le buffer.

### 6.4.3 Une méthode expérimentale pour le calcul du délai

Dans le réseau filaire, plusieurs implémentations de TCP effectuent des mesures du RTT (*Round Trip Time*) qui est l'intervalle de temps entre l'émission d'un segment et la réception d'un acquittement de ce segment. Un mécanisme de mesure de RTT qui utilise une nouvelle option de TCP : *estampille de temps* (TCP timestamps option) a été introduit [111]. Ce mécanisme permet de mesurer les RTT en effectuant la différence entre la valeur de l'estampille envoyée dans un segment et la valeur de l'estampille reçue dans l'acquittement de ce message. Ce mécanisme n'est pas applicable dans un réseau ad hoc. QOLSR calcule le délai sur les liens entre un nœud et ses voisins en utilisant les messages de contrôle *Hello*. Intégrer la valeur de l'estampille dans chaque message *Hello* envoyé ne permet pas de calculer le temps d'aller-retour puisque les messages *Hello* ne sont pas acquittés, ni le temps d'aller simple dû à la désynchronisation des horloges.

Plusieurs travaux de recherche, [112] le plus remarquable, ont été proposés pour étudier le problème de la synchronisation dans le réseau ad hoc. Une idée très simple consiste à faire coexister le protocole de routage avec celui de la synchronisation. Dans ce cas là, le calcul du délai d'aller simple devient facile. Malheureusement, cette solution est coûteuse en terme de bande passante et les horloges des mobiles risquent de ne pas converger.

Nous proposons un algorithme efficace qui calcule le délai d'aller-retour via les transmissions et réceptions des messages *Hello* dans un réseau ad hoc avec des horloges asynchrones. Soit  $a$  un nœud et  $b$  son voisin direct avec un lien symétrique (figure 6.10).

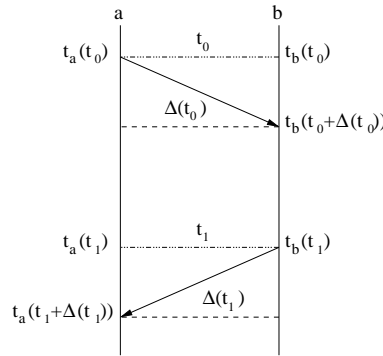


FIG. 6.10 – Le calcul de délai par échange des messages *Hello*.

Soit  $t_0$  le temps absolu lors de la génération d'un message *Hello* par le nœud  $a$ . À ce moment, le temps local affiché par  $a$  est  $t_a(t_0) = t_0 + \delta_a(t_0)$  et  $t_b(t_0) = t_0 + \delta_b(t_0)$  par  $b$  où  $\delta_a(t_0)$  et  $\delta_b(t_0)$  sont les différences entre le temps absolu et les temps locaux indiqués par  $a$  et  $b$  et respectivement. Le nœud  $a$  inclut le temps  $t_a(t_0)$  dans son message *Hello* généré. Lorsque le nœud  $b$  reçoit ce message, son horloge locale indique un temps  $t_b(t_0 + \Delta(t_0)) = t_0 + \delta_b(t_0 + \Delta(t_0)) + \Delta(t_0)$ , il calcule la différence entre son temps local et le temps  $t_a(t_0)$  inclut dans le message *Hello*. Cette différence représente  $\Delta_{\text{par}(b,a)} = \delta_b(t_0 + \Delta(t_0)) + \Delta(t_0) - \delta_a(t_0)$ . Soit  $t_1$  le temps absolu lors de la génération d'un message *Hello* par le nœud  $b$ . À ce temps, les temps locaux affichés par  $a$  et  $b$  sont respectivement  $t_a(t_1) = t_1 + \delta_a(t_1)$  et  $t_b(t_1) = t_1 + \delta_b(t_1)$ . Le nœud  $b$  intègre  $t_b(t_1)$  et son

$\Delta_{\text{pars}(b,a)}$  dans son message *Hello*. Lorsque le nœud  $a$  reçoit le message *Hello* envoyé par  $b$ , son horloge locale indique un temps  $t_a(t_1 + \Delta(t_1)) = t_1 + \delta_a(t_1 + \Delta(t_1)) + \Delta(t_1)$ , il calcule la différence entre son temps local et le temps attaché dans le message reçu  $t_b(t_1)$ , ce qui donne un temps  $\Delta_{\text{pars}(a,b)} = \delta_a(t_1 + \Delta(t_1)) + \Delta(t_1) - \delta_b(t_1)$ .

Maintenant, le nœud  $a$  possède deux valeurs :  $\Delta_{\text{pars}(a,b)}$  et  $\Delta_{\text{pars}(b,a)}$ . En supposant que les horloges locales de  $a$  et  $b$  sont parfaites et sans dérive, on obtient  $\delta_b(t_0 + \Delta(t_0)) = \delta_b(t_1)$  et  $\delta_a(t_0) = \delta_a(t_1 + \Delta(t_1))$ . La somme de  $\Delta_{\text{pars}(a,b)}$  et  $\Delta_{\text{pars}(b,a)}$  donne le délai aller-retour  $\Delta(t_0) + \Delta(t_1)$ .

Afin d'éviter les variations des délais aller-retour mesurés dans MANET, chaque nœud doit étaler et corriger le délai aller-retour mesuré en l'intégrant avec les anciennes mesures. Le délai moyen corrigé d'aller-retour est calculé de la manière suivante :

$$\text{Délai moyen corrigé A-R} = \alpha \times \text{Délai moyen corrigé A-R} + (1 - \alpha) \times \text{Délai A-R} \quad (6.34)$$

où  $\alpha$  est un facteur de correction.

#### 6.4.4 Une méthode expérimentale pour le calcul de la bande passante

Le problème majeur qui réduit significativement la bande passante d'un réseau MANET est les interférences. La transmission d'un flux dépend fortement des autres transmissions dans le voisinage de l'émetteur et du récepteur. Afin d'assurer une garantie de service en terme de bande passante, le modèle d'interférence doit être pris en considération. Nous avons modélisé les interférences par un graphe de conflit.

##### Les interférences

Les transmissions radio génèrent deux zones : la portée de communication ( $R_{tx}$  : transmission range), la portée de détection ( $R_{cs}$  : carrier sensing range) (voir figure 6.11).

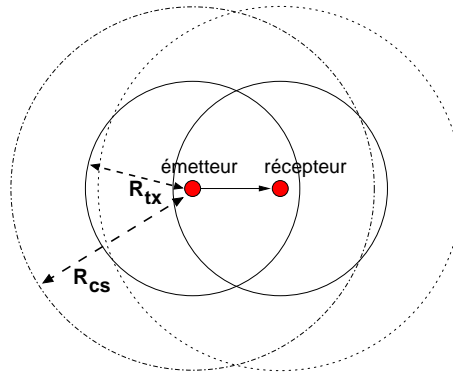


FIG. 6.11 – Les zones d'une transmission radio.

- la portée de communication ( $R_{tx}$ ) : représente la zone où tout signal est parfaitement traité et compris. Sans interférence, la réception d'un paquet est toujours avec succès. La portée de communication est déterminée par la puissance de transmission et les propriétés de propagation des signaux radios.
- la portée de détection ( $R_{cs}$ ) : constitue l'espace où un nœud détecte de l'activité mais n'est pas en mesure d'en tirer de l'information.

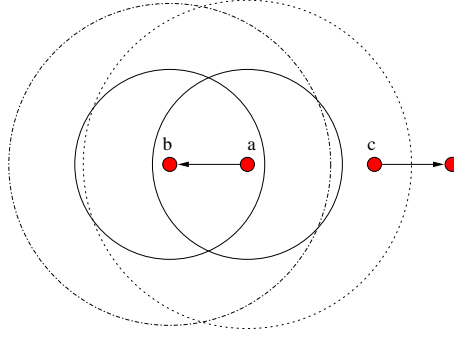


FIG. 6.12 – Une situation délicate.

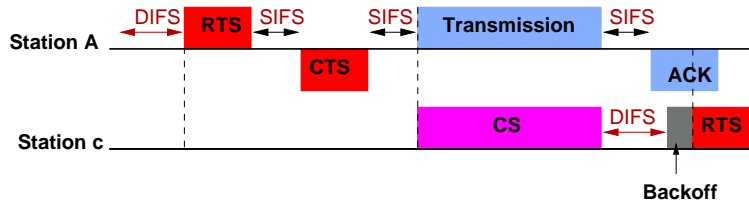


FIG. 6.13 – Chronogramme associé au scénario de la figure 6.12.

Les nœuds qui se trouvent hors la portée de détection de l'émetteur mais dans la portée de détection du récepteur sont appelés des nœuds cachés. Lorsqu'un nœud détecte de l'activité dans son  $R_{cs}$ , il considère le médium occupé. Il se place alors en mode de détection de porteuse (CS : Carrier Sense) avant de reprendre son cycle de transmission. Dans la figure 6.12, le nœud  $c$  se trouve dans cette situation : il ne détecte que du bruit de la part de  $a$ . Par contre il est trop éloigné de  $b$  pour le détecter. Or lorsque  $c$  se trouve dans la zone  $R_{cs}$  de l'émetteur, il n'a pas pu, à partir du RTS ou du DATA, conclure la valeur du champ Duration : il se positionne donc en CS et non en DIFFER (rester inactive), ce que l'on aperçoit sur la figure 6.13. On a montré ici le cas où  $c$  a réussi à débiter sa transmission (le médium lui apparaît en effet libre) alors que  $b$  émettait son Acquiescement. On a donc une collision au nœud  $a$ , en supposant que le signal de  $c$  perçu par  $a$  est bien sûr plus faible que celui perçu de  $b$  mais suffisant pour détériorer le paquet d'ACK. Il s'agit ici d'une faille qui oblige la norme à inclure un nouveau délai d'attente après chaque période CS, équivalent à  $SIFS+ACK+DIFS$ . Ce délai qui est nommé EIFS (*Extended Inter Frame Space*), permet la bonne réception des ACKs (ici en  $a$ ) et de reporter le début du décompte du Backoff au même moment pour les deux nœuds concernés ( $a$  et  $c$ ), afin que la prise du médium reste équitable.

### Les interférences inter et intra-flux

Les interférences inter-flux sont dues au passage de deux ou plusieurs flux dans la même région de voisinage. Ce problème peut être résolu en utilisant un modèle d'interférence efficace, un contrôle d'admission et réservation. Les interférences intra-flux sont dues au passage des paquets de même flux sur le chemin vers la destination puisque les liens ne peuvent pas tous être actifs pour relayer ce flux à la fois.

Soit le réseau montré dans la figure 6.14, où le nœud  $a$  représente la source et  $f$  la destination. Supposons pour le moment que les transmissions radios des nœuds qui ne sont pas voisins ne s'interfèrent pas. Les nœuds  $a$  et  $b$  ne peuvent pas transmettre simultanément puisque le nœud  $b$

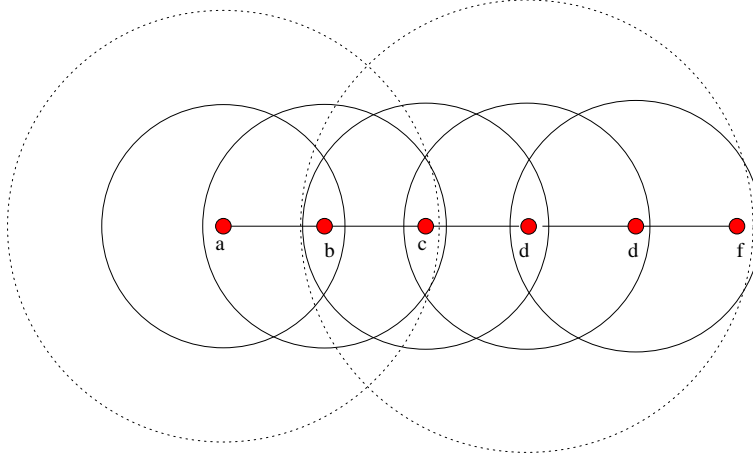


FIG. 6.14 – Un exemple d'une topologie MANET.

ne peut pas recevoir et transmettre en même temps. Les nœuds  $a$  et  $c$  ne peuvent pas transmettre simultanément puisque le nœud  $b$  ne peut pas écouter correctement la transmission de  $a$  si  $c$  transmet en même temps. Par contre, les nœuds  $a$  et  $d$  peuvent transmettre simultanément. Ceci conduit à une utilisation du canal de  $\frac{1}{3}$ , et donc la capacité de chemin est de  $\frac{\min\{Bw_{ab}, Bw_{bc}, Bw_{cd}, Bw_{de}, Bw_{ef}\}}{3}$ .

Cependant, si on suppose que les transmissions radio peuvent s'interférer au delà de  $R_{tx}$ , c'est-à-dire, dans toute la zone  $R_{cs}$ , la situation devient pire. D'après la figure 6.14, la transmission des paquets par le nœud  $d$  interfère avec les paquets ou RTS envoyés par le nœud  $a$  au nœud  $b$ . Dans ce cas le taux maximum d'utilisation de la routes est de  $\frac{1}{4}$ .

La bande passante disponible d'un chemin trouvé par le protocole QOLSR ne prend pas en considération les interférences intra-flux. En effet, si la distance entre la source et la destination est 1-saut, alors la bande passante disponible de ce chemin est exactement la bande passante disponible de lien. Si la distance source-destination est de deux sauts, alors la bande passante maximale disponible de ce chemin est égale à la bande passante trouvée par QOLSR divisée par 2. Si la distance source-destination est de 3 sauts, la bande passante maximale disponible de ce chemin est égale à la bande passante trouvée par QOLSR divisée par 3. Si la distance source-destination est supérieure à 3 sauts, la bande passante maximale disponible de ce chemin est égale à la bande passante trouvée par QOLSR divisée par 4.

### Le graphe de conflit des liens

Nous considérons un réseau sans fil ad hoc de  $n$  stations. Chaque station possède une carte sans fil Wi-Fi. En utilisant la couche MAC IEEE 802.11, des collisions peuvent facilement apparaître lors de transmission et/ou réception. Soit  $i$  un nœud émetteur et  $j$  sont voisin. Si  $i$  utilise le mécanisme de base d'accès au support, une collision est possible : 1) lors de la transmission de son paquet de données avec une transmission simultanée d'un autre nœud (émetteur ou récepteur) appartenant au  $R_{cs}$  de  $i$ , 2) lors de la réception de ce paquet par  $j$  avec une transmission simultanée d'un autre nœud (émetteur ou récepteur) appartenant au  $R_{cs}$  de  $j$ , 3) lors de la transmission de l'ACK par  $j$  avec une transmission simultanée d'un autre nœud (émetteur ou récepteur) appartenant au  $R_{cs}$  de  $j$ , 4) lors de la réception de l'ACK par  $i$  avec une transmission simultanée d'un autre nouveau nœud (la mobilité) appartenant au  $R_{cs}$  de  $i$ . Les mêmes collisions peuvent se produire avec le mécanisme RTS/CTS (RTS au lieu du paquet de données et CTS au lieu de l'ACK).

Les conditions strictes pour avoir une transmission réussie entre  $i$  et  $j$  sont :

$$\text{dist}_{ij} < R_{tx} \wedge \text{dist}_{ik} > R_{cs} \wedge \text{dist}_{jk} > R_{cs}, \quad (6.35)$$

pour n'importe quelle station émettrice ou réceptrice  $k$ .  $\text{dist}_{ij}$  représente la distance entre  $i$  et  $j$ . Ces conditions sont directement reliées aux caractéristiques de la couche physique. En effet, la transmission de  $i$  à  $j$  est réussie si le rapport signal sur bruit au niveau de  $j$  ( $\text{SNR}_{ij}$ ) est supérieur à un seuil :  $\text{SNR}_{ij} \geq \text{SNR}_{\text{thresh}}$ .

Le graphe de connexion  $G$  est un graphe dont les nœuds représentent les stations et les arêtes sont les liens sans fil. Une condition nécessaire d'existence d'arête entre  $i$  et  $j$  si  $\text{dist}_{ij} < R_{tx}$ . Les figures 6.15.a et 6.15.b montrent deux graphes de connexion. Un graphe de conflit des liens (CGL) exprime les interférences entre les liens d'un graphe de connexion. Chaque lien dans  $G$  est représenté par un nœud dans CGL. Deux nœuds dans CGL sont connectés par une arête si les liens correspondants dans  $G$  ne peuvent pas faire passer des flux unicast simultanément.

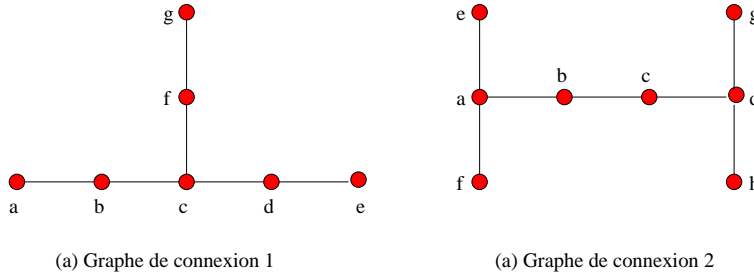


FIG. 6.15 – Deux graphes topologiques.

Pour un lien  $(i, j)$  dans le graphe de connexion, le nœud  $i$  essaie de trouver l'ensemble des liens qui s'interfèrent avec  $(i, j)$  en se basant sur la règle : ***tout lien de distance inférieure ou égale à  $H$  de  $(i, j)$  est un lien d'interférence***. Cette règle est appelée *modèle d'interférence de distance- $H$* . Les nœuds qui possèdent des liens appartenant à l'ensemble des liens d'interférences forment l'ensemble des nœuds d'interférence. En d'autres termes, pour le lien  $(i, j)$ , le nœud  $i$  considère tous les nœuds qui se trouve dans les  $H$ -sauts de  $i$  ou  $j$  comme des nœuds d'interférence.

Les figures 6.16.a et 6.16.b montrent les graphes de conflit des liens correspondants au graphe topologique de la figure 6.15.a construits sur la base des modèles d'interférence de distance-2 et de distance-3 respectivement. Dans la figure 6.16.a, il existe trois cliques (sous graphe complet maximal). Les liens  $\{2, 4, 5\}$  forment l'ensemble des liens d'interférence du lien 1  $= (a, b)$ . Le nœud  $a$  considère l'ensemble  $\{b, c, d, f\}$  comme des nœuds d'interférence avec le lien 1  $= (a, b)$ . Les graphes de conflit des liens correspondants au graphe topologique de la figure 6.15.b construits sur la base des modèles d'interférence de distance-2 et de distance-4 respectivement sont montrés dans la figure 6.17. On peut déjà remarquer que le nombre des cliques diminue lorsque la distance- $H$  de modèle d'interférence augmente.

Le modèle d'interférence de distance- $H$  est une méthode pour détecter les interférences en se basant sur la connectivité. Cette méthode doit être la plus réaliste possible afin d'obtenir la meilleure estimation de la bande passante. Le point critique consiste à déterminer **exactement** pour chaque nœud  $i$  son ensemble des nœuds d'interférence. Cet ensemble doit être : 1) complet : il contient tous les nœuds qui peuvent interférer en cas d'activité avec  $i$ , 2) optimal : il ne contient pas de nœuds dont l'activité ne pose aucun problème d'interférence avec  $i$ .

L'ensemble des nœuds d'interférence d'un nœud donné calculé par le modèle d'interférence de distance- $H$  peut ne pas être complet. Ceci peut se produire puisqu'il est possible qu'un nœud

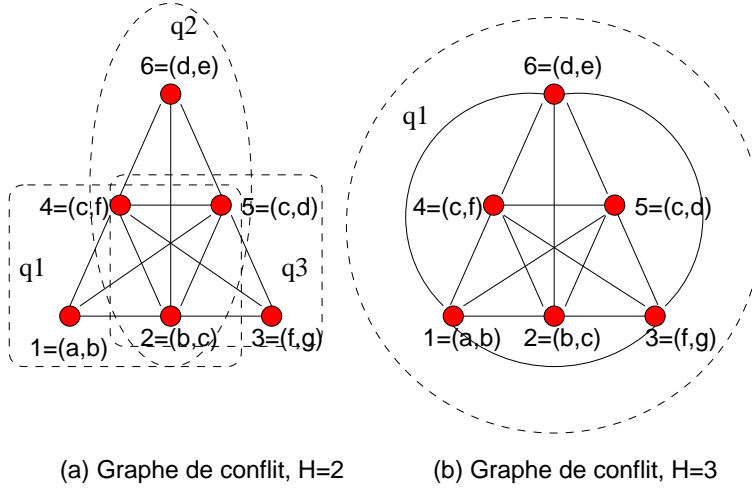


FIG. 6.16 – Les CGL du graphe de la figure 6.15.a.

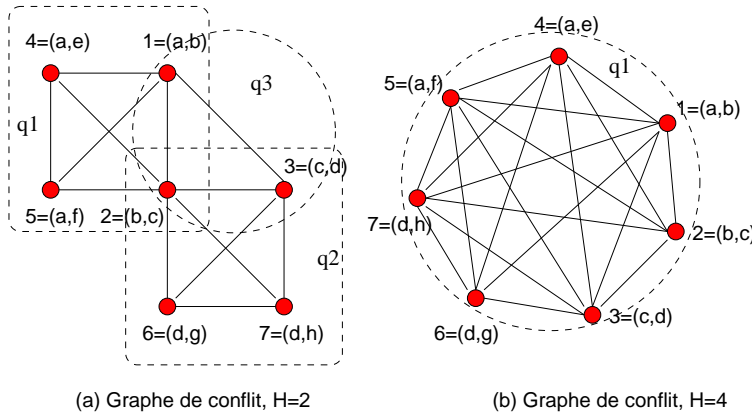


FIG. 6.17 – Les CGL du graphe de la figure 6.15.b.

d'interférence n'est pas dans les H-sauts des deux extrémités d'un lien malgré sa présence dans les  $R_{cs}$  de l'un des deux (voir figure 6.18). Pour résoudre ce problème, une solution consiste à faire augmenter la distance-H de modèle d'interférence. Cette augmentation peut rajouter des contraintes non valides dues aux nœuds qui sont suffisamment loins pour affecter la transmission. Ces contraintes minimisent largement la bande passante disponible calculée. En plus, si un nœud d'interférence n'est même pas connecté au graphe topologique puisqu'il n'appartient à aucune  $R_{tx}$  d'un autre nœud ou il utilise une technologie sans fil différente des autres nœuds (Bluetooth par exemple), il est impossible de détecter ce nœud par la connectivité même si on augmente la distance-H. Une deuxième solution plus efficace, qu'on va montrer par la suite, consiste à ajouter un facteur de correction dans les contraintes de calcul de la bande passante avec une distance-H optimale.

Dans [113], les auteurs ont proposé une solution pour détecter tous les nœuds qui se trouvent dans les zones d'interférences ( $R_{cs}$  de l'émetteur et de récepteur) même s'ils sont isolés sans avoir recours à la connectivité. En effet, chaque station est équipée par un GPS qui calcule sa position par rapport aux stations voisines. Maintenant, il est facile pour une station en utilisant un calcul géométrique de savoir quelles sont les stations qui se trouvent dans sa  $R_{cs}$  et la  $R_{cs}$  de la station

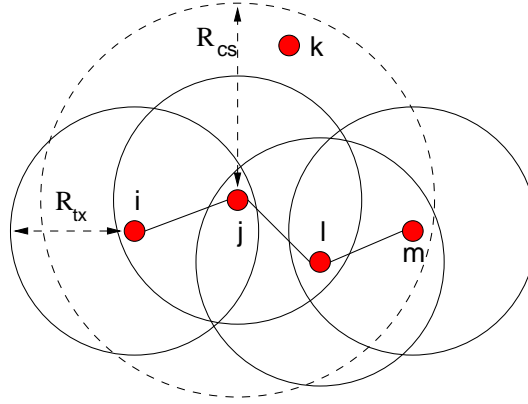


FIG. 6.18 – Il est impossible de détecter le nœud d'interférence  $k$  par  $i$ .

réceptrice et ensuite. Malheureusement, cette solution est coûteuse et ne peut pas être déployer au public pour des raison de sécurité. Un autre problème devient plus évident dans cette solution (utilisation de GPS) que dans notre solution (utilisation de la connectivité), concerne les zones  $R_{tx}$  et  $R_{cs}$ . Ces zones en réalité ne sont pas des cercles parfaits dû au présence des obstacles, fading, etc. Donc, pour une transmission  $i \rightarrow j$ , il est possible que certains nœuds qui se trouvent à une distance inférieure à H-sauts (ou  $R_{cs}$ ) de  $i$  ou  $j$  et qui peuvent transmettre sans interférence avec la transmission de  $i$  comme il est possible que les transmissions de certains nœuds qui se trouvent au delà de H-sauts (ou  $R_{cs}$ ) de  $i$  et  $j$  et s'interfèrent avec la transmission de  $i$ .

### Les contraintes générées par les cliques

Un graphe complet est un graphe dont tous les sommets sont reliés deux à deux. Une clique est un sous-graphe complet qui n'est contenu dans aucun autre sous graphe complet ; par abus de langage, on dénomme souvent par clique tout sous graphe complet. Une clique maximale est une des cliques de plus grande cardinalité parmi toutes celles du graphe. Dans le graphe de conflit de la figure 6.17.a,  $\{1,2,3\}$ ,  $\{1,2,4,5\}$  et  $\{2,3,6,7\}$  représentent des cliques. Les cliques dans un graphe de conflit sont fortement reliées à la capacité du réseau ad hoc. Les nœuds d'une clique dans un CGL représentent l'ensemble maximal des liens en compétition pour accéder au canal. Pas plus d'un seul nœud dans une clique doit être actif à la fois ce qui représente la transition d'un seul flux. Par conséquent, la somme des débits des nœuds formant une clique d'un CGL (ou les flux qui s'interfèrent dans le graphe topologique) ne doit pas dépasser la capacité du canal. On peut mieux comprendre ces contraintes si on représente le débit de chaque flux par un taux d'utilisation du canal. Alors au sein d'une clique ou les flux partagent le canal, la somme des taux ne doit pas dépasser 100%.

Nous considérons le graphe de conflit des liens illustré dans la figure 6.15.a. Soient  $F_1$ ,  $F_2$ , etc, les flux existants respectivement dans les nœuds 1, 2, etc, de CGL. Chaque nœud  $i$  dans la graphe de conflit des liens (ou un lien  $(a,b)$  dans le graphe topologique) a une capacité théorique maximale  $C_i$ . Le système linéaire des contraintes générés par l'ensemble des cliques est :

$$\begin{cases} \frac{F_1}{C_1} + \frac{F_2}{C_2} + \frac{F_3}{C_3} \leq 1 & \text{pour la clique } \{1,2,3\} \\ \frac{F_1}{C_1} + \frac{F_2}{C_2} + \frac{F_4}{C_4} + \frac{F_5}{C_5} \leq 1 & \text{pour la clique } \{1,2,4,5\} \\ \frac{F_2}{C_2} + \frac{F_3}{C_3} + \frac{F_6}{C_6} + \frac{F_7}{C_7} \leq 1 & \text{pour la clique } \{2,3,6,7\} \end{cases}$$

Si on considère que  $C_i = c$  pour tout nœud dans CGL, le système devient plus simple :

$$\begin{cases} F_1 + F_2 + F_3 \leq c & \text{pour la clique } \{1,2,3\} \\ F_1 + F_2 + F_4 + F_5 \leq c & \text{pour la clique } \{1,2,4,5\} \\ F_2 + F_3 + F_6 + F_7 \leq c & \text{pour la clique } \{2,3,6,7\} \end{cases}$$

Ce système linéaire peut être exprimé sous une écriture matricielle puisque le réseau doit satisfaire les contraintes de toutes les cliques. Soit  $Q^i$  la matrice d'incidence d'un nœud  $i$  de CGL d'ordre  $q \times n$ , où  $q$  est le nombre des cliques dans le CGL et  $n$  le nombre des nœuds.  $Q^i$  est définie par :

$$Q_{kj}^i = \begin{cases} 1 & \text{si le lien } (i,j) \in \text{clique } k \\ 0 & \text{si le lien } (i,j) \notin \text{clique } k \end{cases} \quad (6.36)$$

La matrice d'incidence  $Q^i$  de nœud 2 dans le graphe de conflit de la figure 6.17.a est la suivante :

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} q_1 \\ q_1 \\ q_3 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

Soient  $F$  et  $C$  respectivement deux vecteurs-colonnes de dimension  $n$  des flux existants et capacités totales dans chaque nœud de CGL. L'écriture matricielle des contraintes générées par les cliques est :

$$\forall i \quad Q^i F \leq C. \quad (6.37)$$

Les contraintes générées par les cliques présentent les conditions nécessaires pour qu'un réalisable ordonnancement puisse exister mais malheureusement ces contraintes ne sont pas suffisantes. C'est-à-dire, si les contraintes sont vérifiées, on sait théoriquement qu'un ordonnancement est faisable et on peut le réaliser manuellement mais puisque seules les couches MAC des mobiles sont responsables d'ordonner les paquets selon le protocole CSMA/CA, elles peuvent ne pas le trouver.

Les contraintes des cliques sont suffisantes dans le cas où le graphe de conflit des liens est un *graphe parfait* [114].

**Définition 6.4.1** *Le graphe de conflit des liens CGL est parfait si pour tous les sous graphes  $H$  induits de CGL, on a l'égalité entre  $w(H)$  et  $\gamma(H)$ .  $w(H)$  est la cardinalité de la clique maximale du sous graphe  $H$ .  $\gamma(H)$  est le nombre chromatique de  $H$ .*

Comme il est noté dans [113], le *pentagon* (figure 6.19) présente un simple exemple d'un CGL imparfait où les contraintes générées par ses cliques sont insuffisantes. En effet, les contraintes du *pentagon* proposent une capacité de transmission des flux de  $0,5 C$  sur chaque lien, mais en réalité seulement  $0,4 C$  est possible sur chaque lien puisque au plus 2 nœuds du pentagon peuvent être actives simultanément. La figure 6.20 montre l'impossibilité d'ordonner les flux avec une capacité de  $0,5 C$  pour chaque flux sur un temps de 10 unités contre la faisabilité avec  $0,4 C$ . Dans



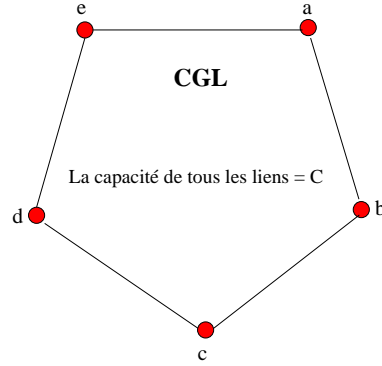
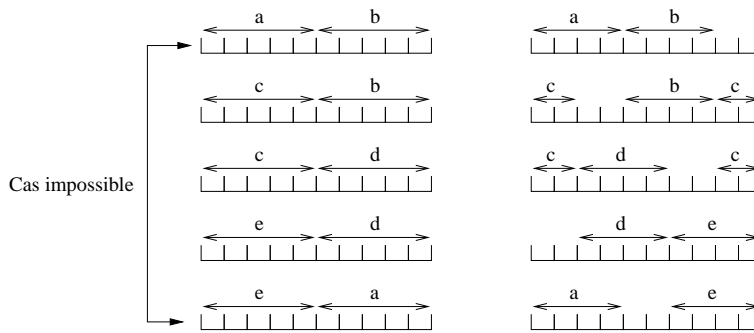


FIG. 6.19 – Un CGL pentagon.

FIG. 6.20 – Ordonnancement des flux avec des capacités de  $0,5 C$  et  $0,4 C$ .

notre cas, il n'y a aucune garantie que le graphe de conflit des liens généré à partir du graphe topologique est parfait et donc on ne sait pas si les contraintes des cliques sont suffisantes ou non.

Les auteurs de [113] ont prouvé que les contraintes générées par les cliques sont aussi suffisantes si elles sont corrigées par un facteur de 0.46. Ces cliques doivent être impérativement extraites d'un graphe de conflit qui est un disque unitaire (UDG : *Unit Disk Graph*) [115]. Les nœuds d'un CGL disque unitaire sont les points au milieu de chaque lien dans le graphe topologique et il existe un lien entre deux nœuds ssi la distance euclidienne entre ces deux nœuds est au plus égale à 1 ou une constante  $R_{cs}$ . Dans notre cas, les nœuds ne possèdent pas la technologie GPS et donc il est impossible de construire un graphe de conflit UDG à partir de graphe topologique ce qui empêche d'utiliser le facteur de correction 0,46 pour rendre les contraintes suffisantes.

La question qui se pose maintenant : **comment obtenir des contraintes suffisantes ?** Une solution possible consiste à vérifier dans un premier temps si le graphe de conflit est parfait. Cette vérification nécessite beaucoup de calcul pour trouver la clique maximale dans chaque sous graphe et elle souffre de problème de passage à l'échelle. Il faut noter que le problème de trouver une clique maximale est un problème NP-complet [116]. Il reste donc à exploiter l'idée d'un facteur de correction rajouté au système linéaire. Ce facteur va nous aider à résoudre deux problèmes en même temps : proposer des contraintes suffisantes et renforcer le modèle d'interférence de distance-H. En effet, notre modèle d'interférence peut négliger des liens dans le calcul comme il peut considérer d'autres liens qui sont totalement indépendants (ne posent pas des problèmes d'interférence) dans le calcul. Par conséquent, le système linéaire peut être sous-contraints et/ou sur-contraints en termes d'absence des flux importants qui doivent figurer dans les équations du

système et/ou la présence de d'autres flux indépendants dans une ou plusieurs équations. Ce facteur, noté  $\beta$  ( $0 < \beta \leq 1$ ), corrige les équations et naturellement il doit être petit si le système est sous-contraint et grand si le système est sur-contraint afin de réduire et augmenter respectivement l'espace des solutions. Nous allons déterminer la valeur de  $\beta$  selon le modèle d'interférence de distance-H.

Maintenant, le système linéaire devient :

$$\forall i \quad Q^i F \leq \beta C. \quad (6.38)$$

### Estimation de la bande passante disponible des liens

D'une manière distribuée, les contraintes générées par les cliques peuvent être utilisées pour estimer la bande passante disponible sur un lien. Puisqu'un nœud dans un CGL peut appartenir à plusieurs cliques, ce nœud prend la bande passante disponible la plus faible parmi celles calculées pour lui dans toutes les cliques. La bande passante disponible d'un nœud  $i$  de CGL est :

$$\begin{aligned} \Gamma^i &= \min\{(C^i \times \beta) - Q^i F^i\} \\ &= \beta c - \max_{k: Q_{kj}^i \neq 0} \{Q_k^i F^i\}. \end{aligned} \quad (6.39)$$

À partir de graphe de conflit des liens de la figure 6.17.a, la bande passante disponible dans le nœud  $2 = (b, c)$  :

$$\Gamma^2 = \min\{[(C^2 \times \beta) - (F_1 + F_2 + F_3)], [(C^2 \times \beta) - (F_1 + F_2 + F_4 + F_5)], [(C^2 \times \beta) - (F_2 + F_3 + F_6 + F_7)]\}.$$

### Considération des flux broadcasts

La formule précédente (6.39) considère seulement les flux unicast. Les flux en broadcast consomment aussi de la bande passante et doivent être pris en compte.

Un flux broadcast s'interfère avec un autre flux broadcast différemment qu'avec un flux unicast. Puisque les flux broadcast sont transmis sans recours aux ACKs ou un mécanisme de RTS/CTS, les liens occupés par un flux broadcast génèrent moins d'interférence par rapport aux liens occupés par des flux unicast. Pour mieux voir les choses, soit l'exemple de la chaîne montré dans la figure 6.14. On ne considère pas les zones  $R_{cs}$ . Si les nœuds  $c$  et  $e$  envoient des flux broadcast en même temps, les liens  $(b, c)$ ,  $(c, d)$ ,  $(d, e)$  et  $(e, f)$  seront occupés et il y'aura une collision au niveau du nœud  $d$  mais sans conséquence majeure en terme de retransmission (les flux broadcast ne sont pas retransmis en cas de collision).

Maintenant, supposons que le nœud  $c$  veut envoyer un flux broadcast. Le nœud  $e$  ne peut pas envoyer un flux unicast au nœud  $d$  puisqu'une collision au niveau de ce dernier aura des conséquences sur la transmission de  $e$ . Par contre, le nœud  $e$  peut transmettre son flux unicast à  $f$  simultanément avec le flux broadcast de  $c$ .

Plus clairement, les flux broadcasts sont associés avec les nœuds et non pas les liens. Par conséquent, les contraintes additionnelles doivent être exprimées en se basant sur les nœuds en conflit. Pour mettre en évidence les nœuds en conflit, nous construisons un graphe de conflit des nœuds GCN. Chaque nœud dans graphe CGN est un nœud dans le graphe topologique. Deux nœuds dans

un CGN sont connectés par une arête si les nœuds correspondants dans le graphe topologique ne peuvent pas faire passer des flux broadcast simultanément. Nous utilisons un modèle d'interférence à distance-K : pour un nœud  $i$ , *tout nœud de distance inférieure ou égale à  $k$  sauts de  $i$  est un nœud d'interférence*. Les cliques de CGN représentent les ensembles des nœuds que leurs flux broadcast partagent la même ressource de capacité. Si on considère seulement les flux broadcast dans un graphe topologique sans les flux unicast, alors la somme des flux broadcast d'une clique de GCN est bornée par la capacité partagée.

Soit  $\mathcal{Q}^i$  la matrice d'incidence de nœud  $i$  dans CGN définie par l'équation suivante :

$$\mathcal{Q}_{kj}^i = \begin{cases} 1 & \text{si les nœuds } i \text{ et } j \in k \\ 0 & \text{sinon} \end{cases} \quad (6.40)$$

Soit  $\mathcal{F}$  le vecteur-colonne des flux broadcasts envoyés par chaque nœud. La forme finale de système linéaire des contraintes qui regroupe les flux unicast et broadcast est donnée par :

$$\forall i \quad \mathcal{Q}^i F^i + \mathcal{Q}^i \mathcal{F} \leq C. \quad (6.41)$$

La bande passante disponible sur un lien  $(i, j)$  de graphe topologique est :

$$\Gamma_{ij} = \beta c - \max_{k: \mathcal{Q}_{kj}^i \neq 0} \{ \mathcal{Q}_k^i F \} - \max_{k, l \in \{i, j\}} \{ \mathcal{Q}_k^l \mathcal{F} \}, \quad (6.42)$$

où  $\mathcal{Q}_k^l$  est le  $k$ -ème vecteur-ligne de  $\mathcal{Q}^l$ .

### Acquérir CGL, CGN dans QOLSR

Chaque nœud doit être capable de construire les deux graphes CGL et CGN en se basant sur les modèles d'interférence de distance-H et de distance-k respectivement. On suppose que  $k \leq H$ . Selon les deux modèles d'interférence, un nœud doit avoir ses voisins à 1, 2, ..., (H+1)-sauts pour construire son graphe CGL et ses voisins à 1, 2, ..., K-sauts pour construire son graphe CGN. Puisque  $K \leq H$ , chaque nœud doit détecter ses voisins jusqu'à (H+1)-sauts. Par exemple, si  $H = 1$ , la base d'information de voisinage doit contenir les voisins à 1 et 2-sauts. Dans ce cas, les messages *Hello* avec un TTL=1 sont suffisants pour découvrir les voisins à 1 et 2-sauts. Si  $H = 2$ , les voisins à 1, 2 et 3-sauts sont nécessaires et des messages *Hello* avec TTL=2 (les messages sont relayés une fois) peuvent accomplir la tâche. Les messages *Hello* avec un TTL>1, sont relayés (TTL-1) fois uniquement par les nœuds MPRs-2 pour réduire l'overhead d'inondation.

Les flux unicasts existants sur chacun des liens reliant les voisins directs (à 1-saut) doivent être échangés dans les messages *Hello*. Pour obtenir cette information, chaque nœud doit compter pour chacun de ses voisins les flux unicasts (y compris les retransmissions en cas de collision) qui lui sont adressés par ces derniers et ensuite il intègre la quantité de ses flux sur chacun des liens de ces voisins dans le message *Hello*. Puisque les messages broadcasts ne sont pas retransmis en cas de collision, le nœud émetteur peut compter ses flux broadcasts vers ses voisins et intégrer le résultat dans ses messages *Hello*. À noter que les flux unicasts sont déclarés dans le message *Hello* sur chaque lien, tandis que les flux broadcasts sont déclaré pour tous les voisins à la fois.

## 6.5 Contrôle d'admission et de réservation

Le contrôle d'admission et la réservation sont destinées uniquement aux flux QoS et ils sont considérés comme des approches a priori pour réduire l'apparition des congestions. Un contrôle

d'admission permet de vérifier la disponibilité des ressources avant d'accepter un flux QoS. Avec QOLSR, même si le nœud source trouve un chemin pour un flux QoS satisfaisant ses contraintes, ce nœud est incapable de garantir les paramètres de QoS exigés de bout-en-bout et il doit vérifier la validité de ce chemin afin de l'accepter définitivement. Cette vérification est importante puisque le chemin est calculé sur une base de topologie partielle d'une part et pour répondre aux problèmes de congestion et mobilité d'autre part. Ce contrôle d'admission permet de limiter l'apparition des congestions dans le réseau. Identifier, prévoir et réagir face à l'apparition de congestion aura une influence positive sur le bon fonctionnement du réseau et par conséquent sur les délais et les taux de pertes des différents flux.

Un contrôle d'admission peut être effectué uniquement à la source ou d'une manière distribuée à la demande de la source. Le nœud source est capable de prendre une décision sans avoir recours aux nœuds intermédiaires s'il possède la topologie entière du réseau avec les conditions QoS. QOLSR permet à chaque nœud d'acquérir une topologie partielle ce qui est un handicap pour un contrôle d'admission uniquement à la source, contrairement à un contrôle d'admission distribué qui n'a besoin que des topologies locales de voisinage de chaque nœud intermédiaire dans le chemin trouvé par QOLSR. En effet, sur la demande de nœud source, chaque nœud intermédiaire dans le chemin vérifie s'il peut répondre aux besoins de flux QoS et si les contraintes sont vérifiées, il demande au nœud suivant dans le chemin de poursuivre la vérification, sinon, il droppe la demande. Le contrôle d'admission ne se fait pas seulement pour la bande passante, mais aussi pour les autres métriques demandées par le flux QoS.

Une fois un chemin trouvé pour un flux QoS [source, destination, bandwidth, delay, met\_1, ..., met\_n], le nœud source envoie le long de ce chemin une sonde de vérification (CREQ : *Check REQuest*). CREQ contient tous les nœuds de chemin avec les paramètres de QoS exigés par le flux QoS. Nous présentons en détail la procédure d'un contrôle d'admission à base de bande passante disponible et de délai.

Pour vérifier qu'un chemin  $p$  est admissible en terme de bande passante disponible, chaque nœud doit réévaluer les contraintes générées par les cliques de son CGL auquel le lien vers le nœud suivant dans le chemin appartient en intégrant la nouvelle demande du flux QoS. Dans une clique et afin de prendre en considération les interférences intra-flux, le nouveau flux est rajouté à tous les liens appartenant à la fois à la même clique et au chemin  $p$ . Soit  $a$  un nœud et  $b$  son nœud suivant dans le chemin  $p$  avec  $i = (a, b)$ . Soit  $\Gamma_{Q_k}^i$  la bande passante disponible sur le lien  $i$  dans la clique  $Q_k$  et  $Bw(f_p)$  la bande passante demandée par le flux  $f$  sur la chemin  $p$ . Lorsque le nœud  $a$  reçoit la sonde CREQ (ou il génère la sonde s'il est la source), il doit vérifier pour chaque clique  $Q_k$  auquel le lien  $i$  appartient :

$$\sum_{j \in Q_k \cap p} bw(f_p) \leq \Gamma_{Q_k}^i. \quad (6.43)$$

Nous considérons l'exemple présenté dans la figure 6.15.a et son CGL dans la figure 6.17.a. Soit  $a$  le nœud source,  $h$  le nœud destinataire et  $p$  le chemin trouvé par QOLSR entre  $a$  et  $h$  constitué par les liens  $1 = (a, b)$ ,  $2 = (b, c)$ ,  $3 = (c, d)$  et  $7 = (d, h)$ . Soit  $\alpha$  la bande passante demandée par le flux QoS sur le chemin  $p$ . Avant d'envoyer la sonde CREQ, le nœud  $a$  doit évaluer sur le lien  $1 = (a, b)$  les contraintes suivantes :

$$\begin{cases} \alpha_{\text{sur } 1} + \alpha_{\text{sur } 2} + \alpha_{\text{sur } 3} = 3\alpha \leq \Gamma_{q_3}^1 & \text{le lien } 1 \in q_3 \\ \alpha_{\text{sur } 1} + \alpha_{\text{sur } 2} = 2\alpha \leq \Gamma_{q_1}^1 & \text{le lien } 1 \in q_1 \end{cases}$$

Le nœud  $b$  doit évaluer sur le lien  $2 = (b, c)$  les contraintes suivantes :

$$\begin{cases} \alpha_{\text{sur } 1} + \alpha_{\text{sur } 2} + \alpha_{\text{sur } 3} = 3\alpha \leq \Gamma_{q_3}^1 & \text{le lien } 2 \in q_3 \\ \alpha_{\text{sur } 1} + \alpha_{\text{sur } 2} = 2\alpha \leq \Gamma_{q_1}^1 & \text{le lien } 2 \in q_1 \\ \alpha_{\text{sur } 2} + \alpha_{\text{sur } 3} + \alpha_{\text{sur } 7} = 3\alpha \leq \Gamma_{q_2}^1 & \text{le lien } 2 \in q_2 \end{cases}$$

Pour vérifier qu'un chemin  $p$  est admissible en terme de délai, chaque nœud dans le chemin  $p$  doit soustraire son délai sur le lien vers le nœud suivant dans le chemin  $p$  de délai indiqué dans la sonde. Si le délai résultant est supérieur à zéro, ce nœud affiche le nouveau délai dans la sonde CREQ et il le relaie vers le prochain saut, sinon, il détruit la sonde.

Une fois la sonde CREQ arrive à la destination, cette dernière répond par un message unicast RREQ (*Reply REQuest*). Si aucun message CREQ atteint la destination, un Timer (timeout) sera expiré au niveau du nœud source et le flux QoS sera rejeté.

Afin d'éviter le problème de congestion, les nœud intermédiaires doivent considérer dans la phase de contrôle d'admission les bandes passantes demandées par tous les autres sondes acceptées dans le voisinage d'interférence et dont les flux QoS correspondants ne sont pas encore envoyés (CREQs envoyées simultanément), C'est-à-dire, les flux QoS qui sont en phase de contrôle d'admission. Une technique pour faire face à ce problème consiste à utiliser la réservation *soft* qui peut être suivie par une réservation *hard*. Lorsqu'un flux est admis dans un nœud intermédiaire, ce nœud effectue une réservation *soft* avant de renvoyer la sonde. Une nouvelle entrée dans sa table de routage est créée avec une marque softBW et un temps d'expiration. Les ressources réservées en état *soft* peuvent être utilisées temporairement par d'autres trafics (flux best-effort et QoS) mais elles ne peuvent pas être réservées par d'autres flux QoS. Lors d'un contrôle d'admission, les portions des bandes passantes réservées en état *soft* sont aussi considérées dans les contraintes générées par un autre nœud dans le voisinage de conflit avant d'admettre son flux. Par conséquent, les bandes passantes réservées pour les flux QoS et qui sont toujours en état soft-state sont rajoutées aux flux existants et déclarées dans les messages *Hello*. La figure 6.21 présente deux scénarios où une congestion peut se produire après un contrôle d'admission si les nœuds intermédiaires d'un chemin ne prennent pas en compte la bande passante demandée par l'autre flux durant la phase du contrôle d'admission.

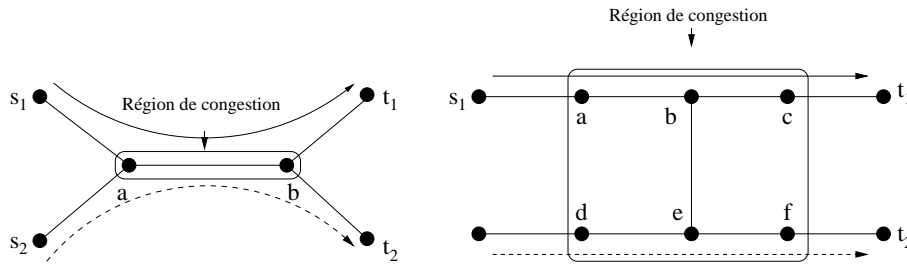


FIG. 6.21 – Scénarios de congestion.

La réception d'un CREQ par la destination signifie que le flux QoS est accepté et une réservation soft est maintenue le long du chemin. La destination répond par un message unicast RREQ qui emprunte le chemin inverse vers la source. Un nœud intermédiaire dans le chemin récepteur d'un RREQ change son état de réservation pour le flux QoS correspondant de *soft* à *hard* en marquant l'entrée dans sa table de routage correspondante par hardBW. La réservation hard permet de garder définitivement les ressources qui sont réservées en soft et utilisées par d'autres trafics pour uniquement le flux QoS correspondant. Une réservation hard est contrôlée par un temps d'expiration

qui doit être rafraîchi périodiquement.

### 6.5.1 Maintenance de la réservation

Une fois le flux de données est transmis entièrement, la source peut libérer explicitement les ressources au moyen d'un message de libération des ressources sur la route. Cependant, ce message peut ne pas atteindre tous les nœuds intermédiaires du chemin de bout-en-bout suite à des problèmes de mobilité des routeurs ou des congestions. Les réservations des flux QoS sont temporaires et peuvent être expirées au bout d'un délai lorsque aucun message de rafraîchissement ou un paquet de flux de QoS de cette réservation n'est reçu. Le nœud source envoie périodiquement des messages de rafraîchissement et chaque nœud intermédiaire récepteur met à jour le temps d'expiration de la réservation correspondante. Cette technique de rafraîchissement permet de ne pas pénaliser les applications ayant un profil d'émission irrégulier. Le rafraîchissement est effectué aussi par les paquets effectifs des flux QoS de la réservation pour faire face au problème de perte des messages explicites. L'accouplement des deux méthodes de rafraîchissement (messages explicites et paquets effectifs) limite la libération des ressources pour un problème temporaire sur le chemin et augmente la stabilité de réseau.

### 6.5.2 La déconnexion des routes

Lorsqu'une route disparaît du fait de la mobilité d'un routeur, les nœuds en aval du point de cassure n'ont plus de paquets à transmettre, qu'il s'agisse de paquets de données effectifs ou de paquets de rafraîchissement de route. En revanche, les routeurs en amont du point de cassure continueront à retransmettre les paquets du flux, ceux-ci étant perdus au niveau du dernier routeur avant le point de cassure. Les messages *TC* précoces préviennent la source de la déconnexion de la route. Lorsque la source se rend compte de la déconnexion d'une route pour un flux QoS (en vérifiant l'existence de la route dans la nouvelle topologie), elle construit une nouvelle route en utilisant le protocole QOLSR et elle envoie aux anciens nœuds intermédiaires un message de libération des ressources. Les routeurs en amont du point de cassure reçoivent le message et ils relâchent les ressources. Les ressources allouées dans les nœuds en aval du point de cassure sont libérées suite à l'absence d'un rafraîchissement de la réservation.

### 6.5.3 Transmission de la signalisation

Nous avons vu que des sondes en mode unicast sont rajoutées pour procéder à un contrôle d'admission et réservation, rafraîchissement et libération des ressources. Afin de limiter la consommation de la bande passante, ces messages peuvent être envoyés dans un paquet QOLSR en même temps qu'avec les messages *Hello* et *TC* qui sont périodiques. Puisque les paquets de QOLSR sont en mode broadcast, un traitement spécial dans chaque nœud est nécessaire pour extraire les sondes et les relayer dans un autre paquet QOLSR si le nœud récepteur est un MPR.

## 6.6 Contrôle de congestion

Dans un réseau ad hoc, le contrôle d'admission et la réservation ne peuvent pas garantir une QoS de bout-en-bout de moment où la topologie et les conditions QoS peuvent changer après l'admission d'un flux. En effet, les congestions dans MANET peuvent apparaître fréquemment

sous la mobilité. Un mécanisme de contrôle de congestion est nécessaire pour identifier et réagir face aux régions de congestion.

Le contrôle de congestion consiste en deux algorithmes : la gestion des files et l'ordonnement. La gestion des files est nécessaire pour équilibrer la trafic. Elle évite la monopolisation de la queue par un seul flux. Une simple gestion de file n'évite pas que les queues soient pleines pour des périodes longues, alors que pour avoir de faibles délais il est souhaitable que les queues ne soient pas trop chargées. En effet, de petites files d'attente réduisent les délais de transmission. L'objectif de la bufférisation dans le réseau est avant tout d'absorber des pointes de trafics fugitives. Pour ces raisons des mécanismes supplémentaires complètent la simple gestion de file d'attente de sortie. L'objectif de ces mécanismes est de diminuer le nombre de paquets éliminés, de diminuer le délai de bout-en-bout, d'éviter le remplissage permanent des files d'attente et cela tout en gardant une bonne utilisation du réseau.

L'ordonnement détermine quel est le prochain paquet à envoyer sur le lien selon la priorité des paquets, les pertes des paquets dues aux congestions et la taille des files d'attente.

L'ordonnement détermine quel est le prochain paquet à envoyer sur le lien selon la priorité des paquets, les pertes des paquets dues aux congestions et la taille des files d'attente.

Les travaux sur les techniques d'ordonnement ainsi la détection de congestion sont en cours. Les probabilités de perte dues aux congestions sont incluses dans les messages *TC*. En se basant sur ces derniers, le nœud source peut détecter facilement si un de ses nœuds intermédiaires est congestionné ou non. Si le problème persiste sur la route au-delà d'un certain temps, la source décide de libérer ses ressources réservées sur le chemin et procède à un recalcul de route.

## 6.7 QOLSR : Évaluation des performances

Dans cette section nous étudions les performances du protocole QOLSR. Cette étude est basée sur des simulations intensives. Le réseau ad hoc est modélisé par un graphe aléatoire avec des nœuds fixes ou mobiles. Le module QOLSR est testé à grande échelle puisque le nombre de nœuds peut dépasser 100000 nœuds.

Le modèle de simulation est basé sur le simulateur OPNET [41]. Ce simulateur offre des outils et des structures hiérarchiques pour simuler, déboguer et analyser des projets de simulation en toute simplicité et efficacité.

Les performances du modèle d'interférence, les contraintes générées par les cliques du graphe CGL, les heuristiques de sélection des nœuds relayeurs et le comportement du protocole QOLSR avec ses heuristiques sont analysées. Ces performances sont exprimées par le taux de délivrance, le délai moyen de bout-en-bout, la différence par rapport aux routes optimales, l'overhead, etc. Plusieurs scénarios sont considérés : un réseau ad hoc fixe ou dynamique, une mobilité faible ou rapide, une charge du trafic faible ou forte, une région de déplacement des nœuds de petite ou large taille, la densité du trafic, etc. Des comparaisons sont effectuées le long de ce chapitre entre le protocole OLSR, le protocole QOLSR utilisant la technique des MPRs classiques (QOLSR\_MPR), QOLSR utilisant la technique des MPRs-1 (QOLSR\_MPRs-1) et QOLSR utilisant des nœuds MPRs-2 (QOLSR\_MPRs-2).

Nous commençons d'abord par une présentation détaillée de notre modèle de simulation. Ensuite, nous présentons les principaux résultats obtenus.

### 6.7.1 Les hypothèses de la simulation

Notre modèle de simulation est basé essentiellement sur les principes suivants :

- Le réseau est représenté par un graphe aléatoire. Dans ce graphe, chaque nœud est placé initialement d'une manière aléatoire et uniforme dans une région déterminée au préalable ;
- Tous les nœuds mettent en œuvre le même protocole de routage avec QoS (QOLSR\_MPRs, QOLSR\_MPRs-1 ou QOLSR\_MPRs-2) ;
- Tous les nœuds sont identiques (les mêmes pouvoirs) mais ils fonctionnent de manière indépendante ;
- Si les nœuds sont mobiles, chaque nœud décide seul de ses mouvements : sa vitesse et sa direction. Les séries successives de vitesses et directions ne sont pas corrélées ;
- L'accès au support de transmission se fait par le protocole CSMA/CA ;
- La portée de communication  $R_{tx}$  est considérée comme un cercle parfait ;
- La région de déplacement des mobiles ne contient aucun obstacle ;

### 6.7.2 Architecture générale

L'architecture générale de notre modèle de simulation est simple, innovante et efficace. Elle consiste uniquement en trois modules qui présentent le fonctionnement de tous les nœuds du réseau. La figure 6.22 montre l'ensemble des modules de cette architecture. Ces modules sont :

- **le module source** : il génère les paquets de données selon une loi de poisson. L'émetteur, le récepteur et la taille des paquets sont sélectionnés aléatoirement.
- **le module routage avec QoS** : il met en œuvre le protocole QOLSR. Il mesure aussi les paramètres de QoS pour répondre aux besoins d'un flux.
- **le module MAC et physique** : il gère les tâches suivantes : 1) la transmission des paquets selon le protocole CSMA/CA ; 2) la propagation du signal sur le canal physique ; 3) la topologie et la mobilité de nœuds.

### 6.7.3 Modèle de simulation

#### Le modèle de la topologie

La topologie du réseau est générée par une distribution aléatoire des nœuds dans une région fixée préalablement par la simulation. Cette région est identifiée par deux valeurs maximales de l'axe des abscisses et des ordonnées ( $area\_max_x$ ,  $area\_max_y$ ). Le nombre de mobiles dans le réseau est un paramètre de simulation. Chaque mobile est représenté par une file d'attente et placé aléatoirement et uniformément dans la région du déplacement. La technique de représenter chaque mobile par une file d'attente permet de réduire le nombre des événements. Cette réduction a un impact sur le temps de simulation. En effet, réduire le nombre des événements amène à une réduction du temps de simulation et une augmentation du nombre de mobiles. Par conséquent, un réseau de 100000 mobiles peut être simulé facilement. Il n'y a pas de transmissions réelles entre les mobiles. Une transmission réussie consiste à enlever le paquet de sa file d'attente source et le mettre dans la file d'attente destinataire.



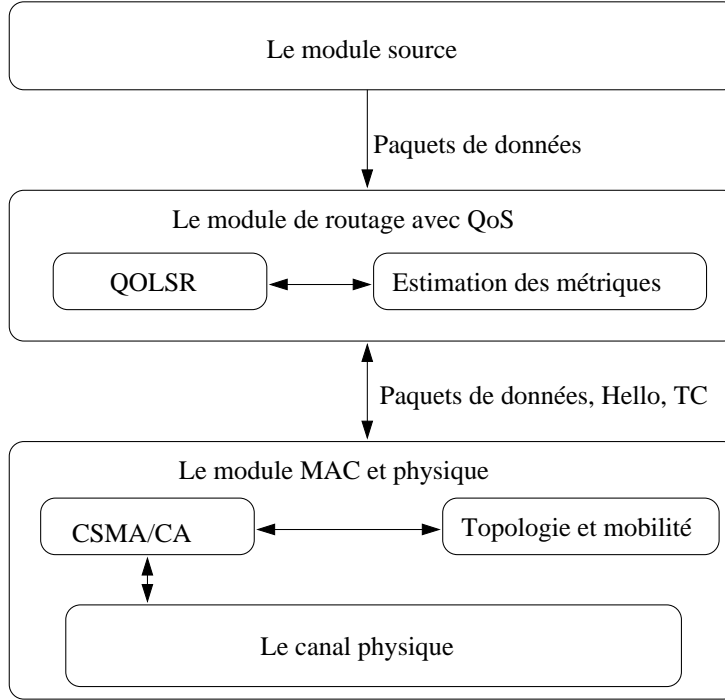


FIG. 6.22 – L'architecture générale.

### Le modèle de mobilité

Le modèle de mobilité est décrit dans le chapitre 3, section 3.10.1.

### Le modèle de trafic

Les paquets de données sont générés selon une loi de poisson de paramètre  $\lambda = \frac{1}{\mu}$  où  $\mu$  est l'intervalle de temps moyen séparant deux générations successives. Chaque nœud dans le réseau est attribué un paramètre de génération  $\lambda_i$  tel que  $\sum_{i=1}^n \lambda_i = \lambda$  où  $n$  est le nombre de nœuds dans le réseau.

Un nœud source sélectionne un nœud destinataire aléatoirement selon une loi uniforme entre 1 et  $n$  afin d'envoyer des paquets de données avec une tailles comprise entre [size\_min, size\_max].

Les messages *Hello*s sont générés périodiquement à chaque Hello\_interval. Les messages *TC* sont générés à chaque TC\_interval, mais aussi à chaque changement dans la liste des MPRs sélecteurs ou les conditions de QoS. Ensuite, ces messages sont envoyés à la couche MAC et physique.

### Le modèle de la propagation radio

La principale supposition de notre modèle de la couche physique est que les signaux envoyés par les émetteurs sont linéairement superposés. Naturellement, nous avons besoin de la puissance d'un signal  $cs_{i,j}$  envoyé par  $j$  au nœud  $i$ . Cette puissance est mesurée par le nœud receptr.

La puissance d'un signal  $Pow(i)$  reçu par  $i$  est  $Pow(i) = \sum_{j=1}^N a_j cs_{i,j}$  où  $a_j = 1$  si  $j$  est un émetteur ou  $a_j = 0$  sinon.

Une simple loi de propagation des signaux radio donne l'expression  $cs_{i,j} = \frac{P_j}{r_{i,j}^\alpha}$  où :

- $P_j$  représente la puissance envoyée par le nœud  $j$  ;

- $r_{i,j}$  représente la distance entre le nœud  $i$  et le nœud  $j$  ;
- $\alpha$  représente le facteur d'atténuation, toujours  $2 \leq \alpha \leq 6$ .

Si le paramètre carrier sensing  $cs_{i,j}$  est supérieur à un certain seuil, le canal est occupé. Il nous reste qu'à préciser les conditions de la bonne réception des paquets. Pour cela, on a supposé qu'un paquet envoyé par un nœud  $j$  à un nœud  $i$  dans un intervalle de transmission  $[t_b, t_e]$  est correctement reçu par le nœud  $i$  si :

- $\forall t \in [t_b, t_e] \quad cs_{i,j}(t) \geq datalevel.$
- $\forall t \in [t_b, t_e] \quad \frac{cs_{i,j}}{\sum_{k \neq j} a_k(t) cs_{i,k}(t)} \geq capturelevel.$

Il y'a trois paramètres : le *datalevel*, le *carrier senselevel* et le *capturelevel*. Le *datalevel* correspond à la puissance nécessaire d'un signal pour qu'il soit bien transmis. Il permet de déterminer la portée de communication ( $R_{tx}$ ). Pour un  $\alpha = 2$ , la valeur  $4.10^{-4}$  de *datalevel* correspond à une portée de communication de 50 m. *carrier senselevel* est calculé à partir de la portée de détection ( $R_{cs}$ ). Pour un  $\alpha = 2$ , la valeur  $10^{-4}$  de *datalevel* correspond à une portée de détection de 100 m. Le *capturelevel* correspond à la perturbation minimale d'un signal pour qu'il soit bien décodé.

#### 6.7.4 Résultats de simulation

Plusieurs simulations sous différents scénarios et paramètres sont effectuées pour étudier le comportement de notre protocole. Dans cette section, nous résumons les résultats les plus concluants.

##### Le modèle d'interférence de distance-H

Cette simulation a pour but de trouver la meilleure distance H qui détecte la majorité de nœuds d'interférence. Nous avons considéré un réseau ad hoc fixe sur une région de  $1500 \text{ m}^2$ . Le nombre de nœuds varie de 50 à 1000 nœuds et la portée de communication de 50 m à 300 m. Les résultats sont donnés en nombre moyen de nœuds par catégorie :

- Les nœuds d'interférence (NI) ;
- Les nœuds d'interference détectés par le modèle ;
- Les nœuds de non interférence considérés comme des nœuds d'interférence par le modèle.

La figure 6.23 est obtenue en utilisant le modèle d'interférence de distance-1 (les voisins à 1 et 2-sauts). Pour n'importe quelle portée de communication, le nombre de nœuds d'interférence augmente avec l'augmentation des nœuds dans le réseau : 14 nœuds d'interférence avec un réseau de 50 nœuds contre 300 nœuds d'interférence dans un réseau de 1000 nœuds. Le modèle d'interférence de distance-1 permet respectivement la détection de (2%, 1.34%) , (12.5%, 10.06%), (29.49%, 26.84%), (52.96%, 49.66%), (80.26%, 77.18%) et (98%, 96%) de nœuds d'interférence dans un réseau de (1000, 50) nœuds en utilisant les portées de communication 50 m, 100 m, 150 m, 200 m, 250 m et 300 m. On remarque que le nombre de nœuds d'interférence détectés par ce modèle augmente avec l'augmentation de la portée de communication et/ou le nombre de nœuds dans le réseau. Ceci est expliqué par le fait que le réseau devient dense avec une importante connectivité et petit diamètre. Par conséquent, les nœuds d'interférence sont proches, connectés et formés uniquement par des voisins à 1 et 2-sauts. Avec un réseau de (1000, 50) nœuds et une portée de communication de 300 m, (10%, 6%) de nœuds qui n'interfèrent pas sont considérés

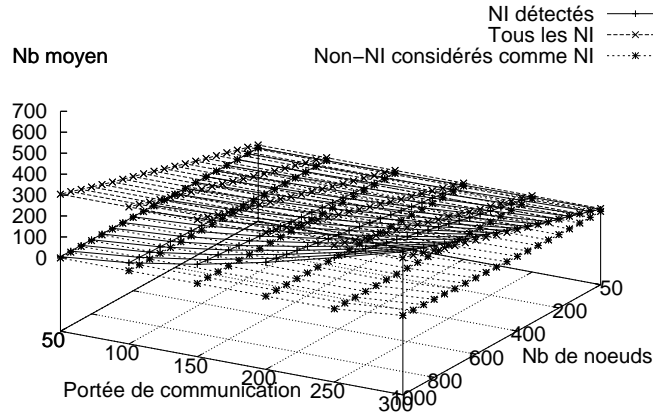


FIG. 6.23 – Les nœuds d'interférence (NI) avec un modèle d'interférence de distance-1.

comme des nœuds d'interférence par le modèle. Le modèle d'interférence de distance-1 est très efficace lorsque le réseau est dense.

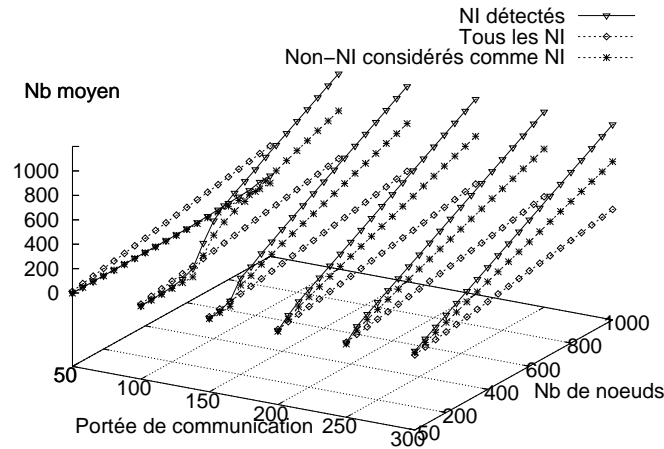


FIG. 6.24 – Les nœuds d'interférence (NI) avec un modèle d'interférence de distance-2.

La figure 6.24 est obtenue en utilisant le modèle d'interférence de distance-2 (les voisins à 1, 2 et 3-sauts). 20% des nœuds d'interférence sont détectés en utilisant 50 m comme une portée de communication. 99% des nœuds d'interférence sont détectés en considérant les portées de communication (100 m, 150 m, 200 m, 250 m et 300 m). 1% des nœuds d'interférence se positionnent au-delà de 3 sauts. Avec ce modèle, 69% des nœuds qui ne sont pas d'interférence mais considérés comme des nœuds d'interférence. Le modèle d'interférence de distance-2 présente la distance minimale pour détecter pratiquement tous les nœuds d'interférence, mais il souffre de la prise en compte de certains nœuds qui n'ont aucun effet d'interférence.

Le modèle d'interférence de distance-2 est le plus intéressant. Il permet de construire un système de contraintes complet en termes de nœuds d'interférence (99%). La présence d'autres contraintes supplémentaires est compensée par l'ajout d'un facteur de correction.

### Le facteur de correction des cliques

Cette simulation a pour but de comprendre l'effet des contraintes générées par les cliques en se basant sur le modèle d'interférence de distance-H. Nous avons pris un réseau fixe de 50 nœuds sur une région de 1500 m<sup>2</sup>. Les liens entre les nœuds sont bidirectionnels. On considère que chaque nœud possède la topologie du réseau et aucune signalisation n'est utilisée. La portée de communication est fixée à 250 m et le portée de détection à 550 m. Chaque nœud possède un buffer MAC de 200 paquets pour un paquet de taille moyenne de 1 Ko. Le débit des paquets de données est augmenté de 100 paquets/seconde (ce qui représente 100 ko par seconde) à 1400 paquets par seconde. Cette charge des paquets utiles est partagée entre tous les nœuds sources (générateurs). On utilise le protocole OLSR classique pour calculer les routes et mesurer la quantité du trafic reçu au niveau de chaque station réceptrice. Le nombre maximum des tentatives de retransmissions est fixé à 7.

Trois probabilités sont utilisées pour choisir les nœuds sources dans le réseau. Avec une probabilité égale à 1 (100%), tous les nœuds du réseau (50 nœuds) génèrent des paquets de données. Dans ce cas, chaque nœud génère en moyenne 2 paquets/s avec une charge de 100 paquets/s dans le réseau et 28 paquets/s avec une charge de 1400 paquets/s. Avec une probabilité de 0.5 (50%), il existe 25 nœuds sources et 10 sources avec une probabilité de 0,2 (20%).

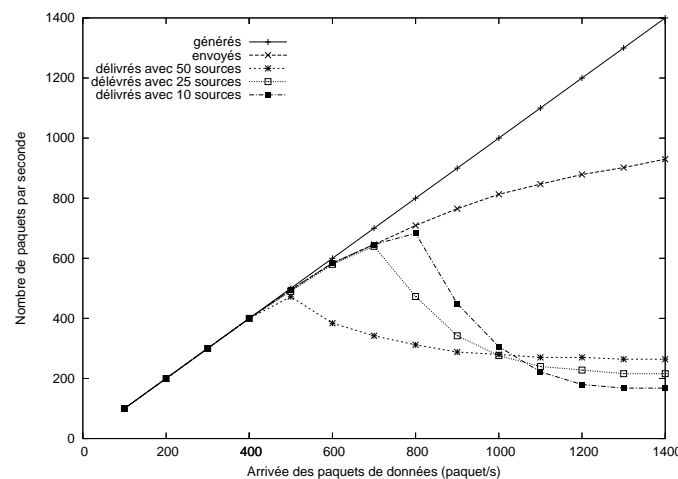


FIG. 6.25 – Le nombre de paquets délivrés par seconde sous différentes valeurs de charge.

La figure 6.25 montre le nombre de paquets délivrés par seconde sous différentes valeurs de charge dans le réseau. Les courbes représentent les paquets générés ainsi les paquets délivrés. On remarque que le protocole fonctionne bien jusqu'à une certaine charge. Après le point de saturation (500 paquets/s pour 50 sources, 700 paquets/s pour 25 sources et 800 paquets/s pour 10 sources), le nombre de paquets délivrés diminue. Ceci est dû à l'absence d'un mécanisme de contrôle de congestion dans OLSR. En effet, lorsqu'un nœud génère des paquets de données avec un débit très élevé, il y'aura de moins en moins de places libres dans son buffer pour router les paquets de données et plus en plus d'interférences dans le réseau. Des paquets sont éliminés pour cause de

buffer saturé ou d'une utilisation d'un nombre limité de tentatives de retransmissions en cas de collision. Nous avons remarqué aussi sans l'indiquer dans les courbes, que les paquets vers des destinations lointaines subissent plus de perte dans les nœuds intermédiaires en cas d'une charge élevée dans le réseau. Le point de saturation est exprimé par une bande passante disponible nulle sur un lien dans le réseau.

En parallèle, lors de l'apparition d'un point de saturation, l'état du réseau est enregistré (la topologie du réseau ainsi les flux sur les liens). En utilisant un programme en le langage C (ce programme est révisé en collaboration avec Ignacy, membre de groupe de recherche QOLSR), chaque nœud calcule la bande passante disponible sur les liens vers ses voisins. En effet, il construit son graphe de conflit de liens CGL selon le modèle d'interférence de distance- $H$  et il génère pour chaque lien le système d'équations linéaires avec la présence du facteur de correction  $\beta$  :

$$\forall i \quad Q^i F \leq \beta C. \quad (6.44)$$

Comme il est montré dans le tableau 6.3, lorsque la distance  $H$  du modèle d'interférence augmente, la charge dans le réseau pour laquelle la capacité du réseau devient nulle (le point de saturation) diminue.

Facteur de correction	Sources	H=1	H=2	H=3
[0, 0,4]	50	[0, 601]	[0, 400]	[0, 213]
	25	[0, 860]	[0, 608]	[0, 421]
	10	[0, 993]	[0, 698]	[0, 557]
]0,4, 0,5]	50	[609, 700]	[405, 589]	[219, 408]
	25	[877, 995]	[617, 783]	[430, 603]
	10	[1006, 1111]	[709, 912]	[566, 713]
]0,5, 1]	50	[700, 1110]	[600, 830]	[411, 515]
	25	[999, 1275]	[794, 1020]	[618, 801]
	10	[1130, 1309]	[925, 1129]	[741, 890]

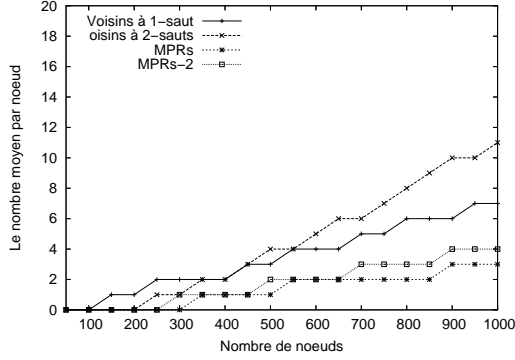
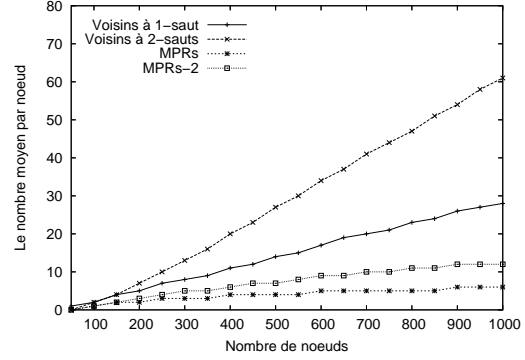
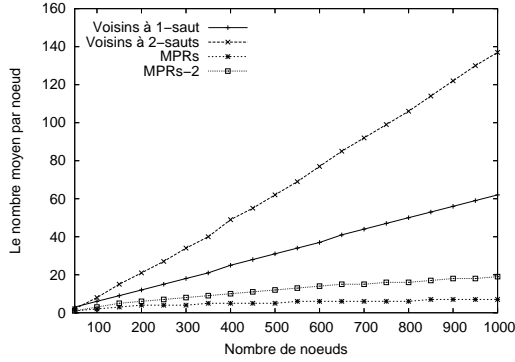
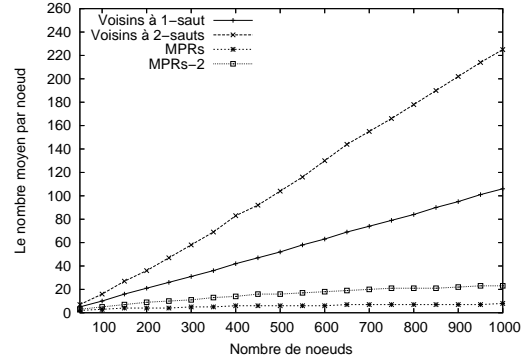
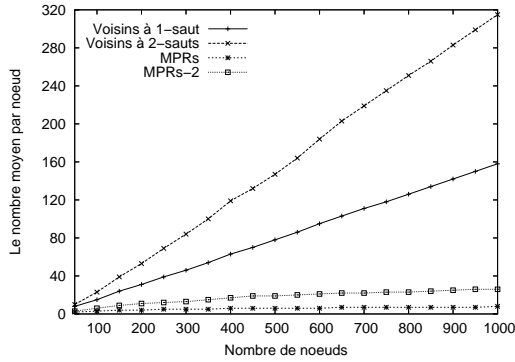
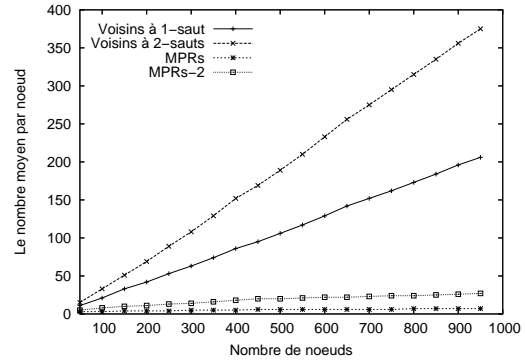
TAB. 6.3 – La limite de la charge (paquets/s) dans le réseau pour obtenir le point de saturation.

Pour 50, 25 et 10 nœuds sources dans le réseau, les points de saturation théoriques sont atteints respectivement à 500 paquets/s, 700 paquets/s, 800 paquets/s. Avec  $H = 1$ , le facteur de correction qui permet d'approcher le point de saturation théorique pour les différents nombre de sources est de  $0,29 \in [0, 0,4]$ . Pour  $H = 2$ , le meilleur facteur de correction est  $0,44 \in ]0,1, 0,5]$  et  $0,71 \in ]0,5, 1]$  pour  $H = 3$ . Le système linéaire de contraintes généré avec un model d'interférence de distance-1 est sous-contraint, une petite valeur du facteur de correction permet de limiter l'espace de solutions pour répondre au manque de certaines contraintes. Lorsque la distance  $H$  augmente, le système linéaire de contraintes devient de plus en plus sur-contraint. Par conséquent, des grandes valeurs de  $\beta$  augmente l'espace des solutions vers des points de saturation proche de ceux dans la réalité.

### Le nombre moyen des MPRs et MPRs-2 par nœud

Dans cette simulation, on s'intéresse uniquement aux MPRs et MPRs-2. Le nombre des MPRs-1 est exactement le même que celui des MPRs classiques. Nous considérons un réseau ad hoc fixe sur une région de  $1000 \times 1000 \text{ m}^2$ . À chaque nœud est assigné un taux de repos du canal

(*idle\_time*) tiré aléatoirement et uniformément entre 0 et 1. Tous les liens entre les nœuds ont une bande passante théorique de 2 Mbps. La bande passante disponible sur un lien entre deux nœud  $a$  et  $b$  est de  $\min(idle\_time_a, idle\_time_b) \times Bw_{ab}$ . Les résultats de simulation représentent les moyennes calculées sur 100 réseaux aléatoires.

FIG. 6.26 –  $1000 \times 1000 \text{ m}^2$  avec  $R_{tx} = 50 \text{ m}$ .FIG. 6.27 –  $1000 \times 1000 \text{ m}^2$  avec  $R_{tx} = 50 \text{ m}$ .FIG. 6.28 –  $1000 \times 1000 \text{ m}^2$  avec  $R_{tx} = 50 \text{ m}$ .FIG. 6.29 –  $1000 \times 1000 \text{ m}^2$  avec  $R_{tx} = 50 \text{ m}$ .FIG. 6.30 –  $1000 \times 1000 \text{ m}^2$  avec  $R_{tx} = 50 \text{ m}$ .FIG. 6.31 –  $1000 \times 1000 \text{ m}^2$  avec  $R_{tx} = 50 \text{ m}$ .

Les figures 6.26, 6.27, 6.28, 6.29, 6.30 et 6.31 obtenues respectivement pour des portées de communication de 50 m, 100 m, 150 m, 200 m 250 m et 300 m illustrent le nombre de voisins à 1-saut, 2-sauts, MPRs et MPRs-2 par nœud en variant le nombre de nœuds dans le réseau. Dans la figure 6.26 (portée de communication de 50 m), le nombre de voisins à 1-saut, 2-sauts, MPRs et MPRs-2 par nœud augmente avec l'augmentation du nombre de nœuds dans le réseau.

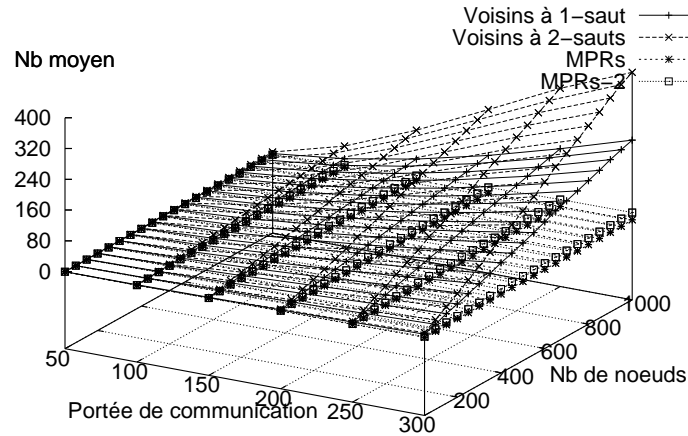


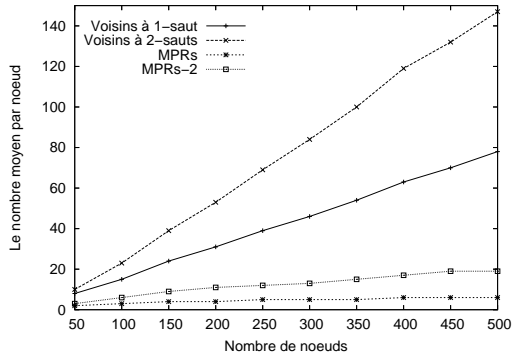
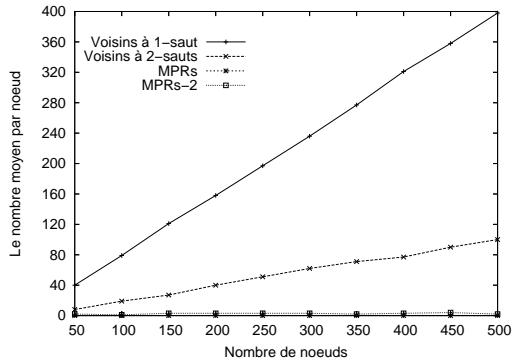
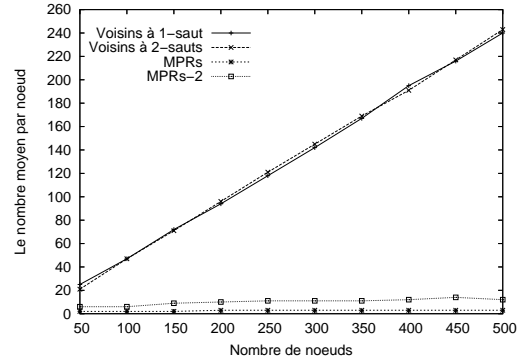
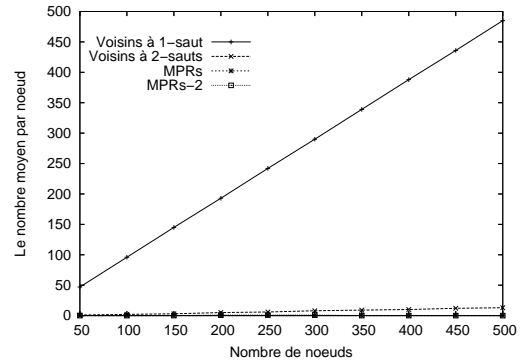
FIG. 6.32 – Le nombre moyen de nœuds dans une région de  $1000 \times 1000 \text{ m}^2$ .

Le nombre de voisins à 1-saut est inférieur au nombre de voisins à 2-sauts : dans un réseau de 1000 nœuds, il existe 7 voisins à 1-saut/nœud contre 11 voisins à 2-sauts/nœud. Il est clair que le nombre de nœuds MPRs et MPRs-2 est inférieur au nombre de voisins à 1-saut puisqu'ils sont des sous ensembles des voisins à 1-saut. Le nombre de nœuds MPRs-2 est supérieur de 12,5% que le nombre de nœuds MPRs. Le même comportement est observé dans le reste des figures sauf pour l'écart entre le nombre de nœuds MPRs-1 et MPRs qui se stabilise à des valeurs inférieures à 40%. La figure 6.32 résume l'ensemble des figures dans un repère de trois dimensions.

Nous considérons maintenant le même réseau mais sur une région de  $200 \times 200 \text{ m}^2$ . À partir des figures 6.33, 6.34, 6.35 et 6.36 on obtient :

- pour une portée de communication fixe, le nombre de voisins à 1, 2-sauts, MPRs et MPRs-2 augmente en fonction du nombre de nœuds dans le réseau ;
- pour un nombre de nœuds fixe dans le réseau, le nombre de voisins à 2-sauts diminue avec l'augmentation de la portée de communication ;
- à partir de la figure 6.34, le nombre de voisins à 2-sauts passe en dessous du nombre de voisins à 1-sauts. Il converge vers la valeur zéro avec l'augmentation de la portée de communication. Ceci est expliqué par le fait que cette dernière couvre toute la région et donc il y a plus de voisins à 1 saut qu'à 2-sauts. Par conséquent, le nombre de nœud MPRs et MPRs-2 diminue sensiblement.
- le nombre de nœuds MPRs-2 par nœud est toujours supérieur à celui des nœuds MPRs.

D'une manière générale, l'écart maximal entre le nombre de nœuds MPRs-2 et celui des MPRs est de 40%. L'écart diminue et les deux nombres convergent vers une même valeur dans un graphe dense.

FIG. 6.33 – 200x200 m<sup>2</sup> avec  $R_{tx} = 50$  m.FIG. 6.35 – 200x200 m<sup>2</sup> avec  $R_{tx} = 150$  m.FIG. 6.34 – 200x200 m<sup>2</sup> avec  $R_{tx} = 100$  m.FIG. 6.36 – 200x200 m<sup>2</sup> avec  $R_{tx} = 200$  m.

### La sélection des nœuds MPRs, MPRs-1 et MPRs-2

Dans cette simulation, nous montrons les performances des routes trouvées par les protocoles : OLSR, QOLSR\_MPRs-1, QOLSR\_MPRs-2 et un algorithme à état de lien. Un réseau fixe de 100 nœuds sur une région de  $1000 \times 1000$  m<sup>2</sup> est considéré. Les résultats sont répertoriés en deux catégories : la performance et le coût. La performance est mesurée par : (i) erreur de débit : le pourcentage des chemins avec une bande passante non optimale ; (ii) la différence moyenne : le pourcentage de différence moyenne entre les bandes passantes optimales et celles trouvées par le protocole de routage (la valeur la plus large représente le plus mauvais résultat). Le coût est mesuré par : (i) overhead : le nombre moyen de messages TC envoyés dans le réseau ; (ii) nombre de relayers : le nombre moyen de nœuds relais dans le réseau. Un nœud sélectionné par plusieurs nœuds comme relayer est compté une seule fois dans cette moyenne.

D'après le tableau 6.4, le protocole OLSR présente les plus mauvaises performances. En effet, il réalise les plus larges valeurs d'erreur de débit et la différence moyenne. Le QOLSR\_MPRs-1 conduit à des meilleures performances que le protocole OLSR puisque les routes sont des chemins shortest-widest. Cependant, les performances des chemins trouvés par QOLSR\_MPRs-1 sont loin des performances des chemins optimaux en termes de bande passante. Ceci est conforme à notre démonstration théorique dans le chapitre précédent. Le protocole QOLSR\_MPRs-2 réalise les meilleures performances. En effet, il trouve des chemins optimaux en termes de bande passante en se basant sur une carte de topologie partielle. Les valeurs d'erreur de débit et la différence moyenne sont nulles.

Le coût est directement lié au nombre de nœuds relayers. En effet, Le coût augmente en



Algorithme	$R_{tx}$	Performance		Coût	
		Erreur de débit	Diff moyenne	Overhead	Nb relais
OLSR classique	300m	28%	46%	12	65
	200m	41%	51%	24	68
	100m	12%	45%	5	42
QOLSR_MPRs-1	300m	14%	22%	12	65
	200m	21%	26%	24	68
	100m	8%	44%	5	42
QOLSR_MPRs-2	300m	0%	0%	26	71
	200m	0%	0%	38	73
	100m	0%	0%	5.7	44
À état de lien	300m	0%	0%	1245	100
	200m	0%	0%	979	100
	100m	0%	0%	28	100

TAB. 6.4 – La performance et le coût.

fonction du nombre de nœuds relayeurs. Le protocole à état de lien génère l'overhead maximal puisque chaque nœud est un relayeur. Les heuristiques de sélection des nœuds MPRs et MPRs-1 réduisent le nombre de nœuds relayeurs. Par conséquent, l'overhead est inférieur à celui dans QOLSR\_MPRs-2.

On remarque que les portées de communication de 100 m et 300 m engendrent moins de nœuds relayeurs qu'une portée de communication de 200 m. Avec 100 m, il existe moins de voisins à 2-saut et donc moins de nœuds relayeurs. Avec 300 m, le réseau devient dense et un petit nombre de nœuds relayeurs peut couvrir un nombre important de voisins à 2-sauts. Avec 200 m, le réseau se trouve entre les deux extrêmes : réseau de faible connectivité et forte connectivité. Ceci conduit à un large nombre de nœuds relayeurs.

### La délivrance des paquets en fonction de la charge du réseau

Nous considérons un réseau ad hoc fixe de 50 nœuds sans liens unidirectionnels. Tous les nœuds sont des générateurs de paquets de données. Chaque nœud possède un buffer MAC de 200 paquets pour un paquet de taille moyenne de 1 Ko. Le débit des paquets de données augmente de 100 paquets/seconde (ce qui représente 100 ko par seconde) à 1400 paquets par seconde. Cette charge des paquets utiles est partagée entre tous les nœuds générateurs. Les courbes de la figure 6.37 représentent le nombre de paquets : 1) envoyés : la première transmission d'un paquet par le nœud source ; 2) retransmis : la retransmission d'un paquet de données ou de contrôle par un nœud intermédiaire, la première transmission n'est pas prise en compte ; 3) délivrés : les paquets de données arrivés à la destination finale.

D'après la figure 6.37, le réseau se comporte bien en termes de délivrance avant le point de saturation (500 paquets/s). Le nombre de paquets délivrés est pratiquement le même pour les protocoles QOLSR\_MPRs ou QOLSR\_MPRs-1. Comme il est expliqué pour la figure 6.25, le nombre de paquets délivrés diminue après le point de saturation. Le protocole QOLSR\_MPRs-2 conserve une stabilité après le point de saturation. Cette stabilité est due à la réaction rapide du protocole au changement de bande passante dans le réseau en calculant d'autres chemins pour contourner les points de congestions. En revanche, ce protocole reste incapable d'éviter l'apparition des conges-

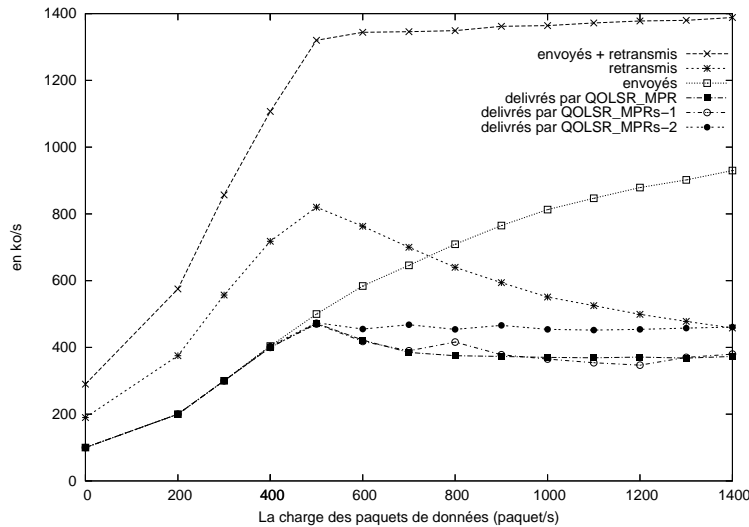


FIG. 6.37 – Le nombre de paquets délivrés par seconde sous différentes valeurs de charge.

tions et de retarder le point de saturation.

### Les performances dans un réseau dynamique

La figure 6.38 montre le pourcentage des paquets délivrés dans un réseau dynamique en utilisant les protocoles OLSR, QOLSR\_MPR, QOLSR\_MPR1 et QOLSR\_MPR2. La vitesse varie de 50 mètres/min (3 km/h) à 500 mètres/min (30 km/h). Le réseau ad hoc comporte 50 nœuds qui se déplacent sur une région de  $1000 \times 1000 \text{ m}^2$ . Tous les nœuds sont des générateurs. La probabilité de déplacement est fixée à 0,2. C'est-à-dire, 20% de nœuds sont mobiles. Chaque nœud mobile maintient sa vitesse et sa direction durant 60 secondes avant de les recalculer à nouveau. À partir de la figure 6.38, le nombre de paquets délivrés diminue en fonction de la vitesse. En effet, chaque nœud maintient son voisin dans sa base d'information de voisinage pendant 6 secondes. Si ce voisin se déplace et qui est le prochain saut dans la transmission, le nœud continue à envoyer ses paquets vers ce voisin. Après un nombre maximal de tentatives de retransmission, il élimine le paquet. Dans ce cas, l'émetteur considère le problème comme une collision et non pas une mobilité de prochain saut. La mobilité peut causer aussi le problème de la non disponibilité d'une route. Par exemple, la destination se trouve isoler temporairement alors qu'un nœud s'apprête à lui envoyer ou relayer des paquets de données. Avec une vitesse de 2 mètres/minute, 99,92% des paquets sont délivrés en utilisant QOLSR\_MPR, 99,01% par QOLSR\_MPR1 et 99,99% par QOLSR\_MPR2. Avec une vitesse de 500 mètres/minute, 88% des paquets sont délivrés par QOLSR, 90,9% par QOLSR\_MPR1 et 97% par QOLSR\_MPR2. Le protocole QOLSR\_MPR2 réalise le meilleur taux de délivrance puisque ses routes sont optimales en termes de bande passante. Les protocoles QOLSR\_MPRs et QOLSR\_MPRs-1 ont le même taux de perte dans le réseau. Il faut noter que les paquets sont aussi perdus pour des problèmes de congestion, collision et la non disponibilité d'une route.

La simulation suivante montre le comportement du protocole QOLSR\_MPR2 face à la mobilité et au BANDWIDTH\_THRESHOLD. Le protocole QOLSR\_MPR2 génère des messages *TC* précoces en cas de changement dans la bande passante au-delà de BANDWIDTH\_THRESHOLD. La figure 6.39 illustre une comparaison entre les taux de délivrance des paquets sous la mobi-

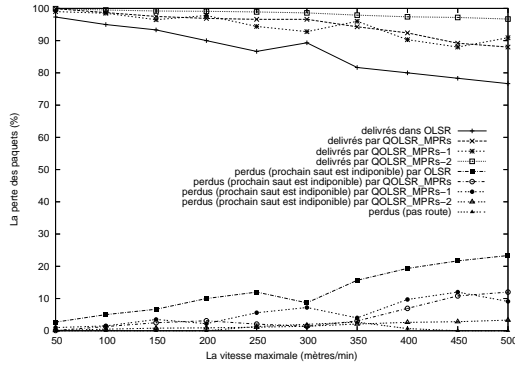


FIG. 6.38 – La délivrance des paquets de données.

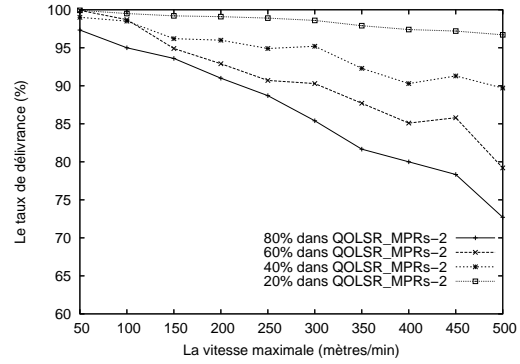


FIG. 6.39 – Le taux de délivrance.

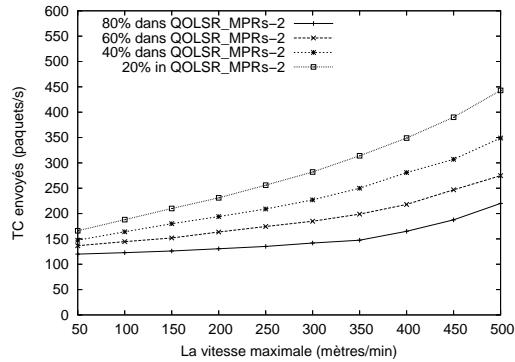


FIG. 6.40 – L'overhead dans le réseau.

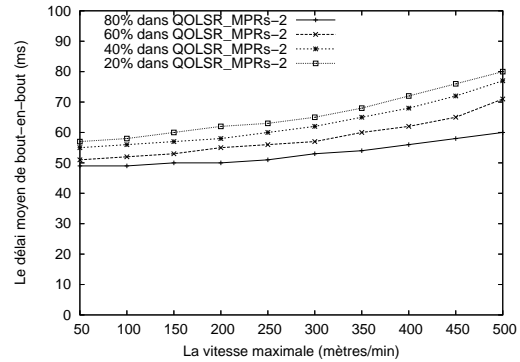


FIG. 6.41 – Le délai moyen de bout-en-bout.

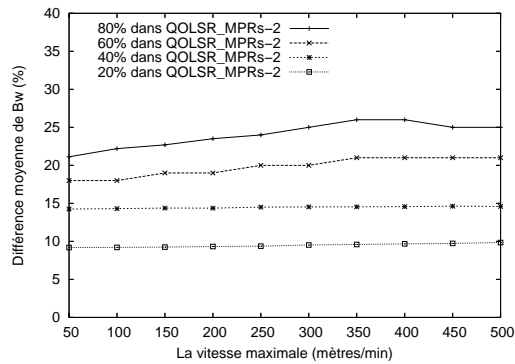


FIG. 6.42 – La différence moyenne de bande passante.

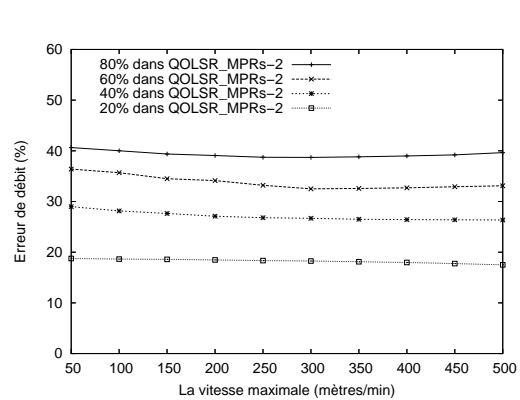


FIG. 6.43 – Erreur de débit.

lité et plusieurs valeurs de BANDWIDTH\_THRESHOLD (20%, 40%, 60%, 80%). Pour une valeur fixe de BANDWIDTH\_THRESHOLD, les paquets arrivent de moins en moins vers leurs destinations lorsque la mobilité augmente. Pour une vitesse constante, le taux de délivrance diminue aussi avec l'augmentation de la valeur de BANDWIDTH\_THRESHOLD. Avec BANDWIDTH\_THRESHOLD = 20%, le protocole QLSR\_MPR2 réalise le meilleur taux de délivrance puisque les nœuds reçoivent d'une manière continue les informations les plus récentes de la topologie et la bande passante du réseau. Ceci permet de maintenir des routes optimales durant toute la

session de transmission. L'augmentation des valeurs de `BANDWIDTH_THRESHOLD` permet de réduire la fréquence de la mise à jour des routes mais en revanche, les routes perdent la garantie d'être optimales.

L'overhead en termes de messages *TC* envoyés dans le réseau en fonction des valeurs de `BANDWIDTH_THRESHOLD` est illustré dans la figure 6.40. Avec `BANDWIDTH_THRESHOLD = 20%`, le réseau génère le plus grand nombre de messages *TC* afin de répondre rapidement au changement de la bande passante. Mais cet overhead a un impact direct sur le délai de bout-en-bout (figure 6.41) qui augmente avec l'augmentation de l'overhead.

Les performances en termes d'erreur de débit et différence moyenne (figures 6.42 et 6.43) donnent avantage au `QOLSR_MPR2` avec `BANDWIDTH_THRESHOLD = 20%`. Il est clair que pour cette valeur l'optimalité des chemins est garantie.

## 6.8 Conclusions

Nous avons présenté un modèle de QoS qui traite l'ensemble des phénomènes qu'une transmission d'un flux peut subir. Ce modèle trouve pour chaque flux une route sur mesure en utilisant un protocole de routage avec QoS. `QOLSR` est conçu d'une manière extensible et il peut respecter n'importe quelle métrique en utilisant des diffusions optimisées et des algorithmes de routage quasi-linaires. Les métriques sont calculées sans aucune information particulière fournie par la couche MAC. Une modélisation efficace des interférences nous a permis de calculer la bande passante disponible sur les liens. `EQMM` est considéré comme une approche à priori pour réduire l'apparition des congestions en utilisant un contrôle d'admission suivi par une réservation. Ces mécanismes sont assurés par une signalisation minimale qui peut être transmise en utilisant les paquets de `QOLSR` déjà existants. Les changements des conditions de QoS du réseau et sa topologie sont gérés par le protocole `QOLSR` contrôle la transmission des flux dans le réseau en respectant les priorités.

Nous avons proposé un modèle de simulation très efficace et optimisé par rapport aux modèles existants pour simuler les performance de `QOLSR`. Ce modèle traduit exactement les opérations réelles qu'un nœud peut effectuer lors de sa présence au sein d'un réseau ad hoc. Des simulations ont été lancées sous plusieurs scénarios et paramètres. Les performances sont étudiées dans un réseau ad hoc fixe et mobile. La simulation des cliques nous a permis de détecter la meilleure valeur pour la distance  $H$  dans le modèle d'interférence. Selon cette valeur, un facteur de correction est positionné. Par exemple, si le modèle d'interférence de distance-2 est utilisé, la meilleure valeur de  $\beta$  est de 0,44. Les différentes heuristiques de sélection des nœuds relayeurs sont analysées. Le protocole `QOLSR_MPRs-2` présente les meilleures performances en termes de bande passante sur les chemins trouvés. En revanche, il génère beaucoup d'overhead qui augmentent le délai de bout-en-bout. L'utilisation des `MPRs-2` s'avère importante dans le cas d'un réseau de grande connectivité. Le protocole `QOLSR_MPRs-2` reste impuissant devant les congestions.



## Chapitre 7

# Conclusions et perspectives

### 7.1 Conclusion

Les réseaux 4G et ad hoc sont actuellement les sujets de recherche les plus actifs. Plusieurs problématiques restent à identifier et élaborer par la suite des solutions de meilleures performances.

Nous avons commencé cette thèse par une étude de différentes technologies de réseaux sans fil et mobiles afin de générer une classification selon l'étendue géographique et la topologie. Cette classification nous a permis de concevoir une structure hiérarchique de réseaux sans fil. Deux concepts sont étudiés : les réseaux cellulaires et les réseaux locaux sans fil. Ces réseaux connaissent actuellement d'importants développements.

Puisque la portée des transmissions radio est limitée, un relaying est nécessaire pour permettre la communication entre un nœud source et un autre destinataire. Plusieurs protocoles de routage ont été proposés dans le groupe MANET. Ils supportent des couches physiques et MAC hétérogènes et offre aux couches supérieures la possibilité de fonctionner uniquement sur IP. Cependant, ces protocoles considèrent uniquement les flux best-effort et ils n'offrent aucune garantie de QoS de bout-en-bout. Cette dernière est le sujet de plusieurs recherches. On trouve parmi les solutions proposées : des modèles de QoS, des protocoles de routage contraints, des protocoles de signalisation, des protocoles MAC avec QoS, etc.

Concernant les réseaux 4G, une structure hiérarchique de plusieurs technologies sans fil (OWN) a été proposée. Cette structure est basée sur la taille de la zone de couverture de chaque technologie sans fil ainsi la bande passante offerte. Chaque niveau hiérarchique est enveloppé dans un autre plus grand et moins performant. Logiquement, le niveau le plus bas dans la OWN est le plus performant en termes de qualité et de débit. Un mobile doit disposer de plusieurs interfaces de technologies sans fil. Il écoute en permanence les signaux pilotes venant des stations de base de différents niveaux. S'attacher à une des stations de base de niveau le plus bas dans la OWN est le but ultime de chaque mobile. Sur ce principe, le handover vertical est conçu. Il permet au mobile de monter et de descendre respectivement dans la structure lorsqu'un certain nombre de signaux pilotes est reçu successivement d'un niveau hiérarchique inférieur dans la OWN ou perdu de niveau hiérarchique courant. Nous avons proposé deux techniques de handovers verticaux qui peuvent réaliser une perte nulle et un temps de latence faible. Il est montré par simulation que les deux techniques réalisent les mêmes performances. La latence du handover vertical représente un obstacle pour les applications temps réels où la gigue doit être minimale. La technique de la cellule virtuelle réduit significativement le temps de latence des handovers verticaux. D'autres stratégies plus agressives ont été proposées mais elles provoquent des overhead conséquents.

Après les handovers verticaux, nous nous sommes intéressés au problème du calcul d'une route dans un graphe pondéré avec plusieurs contraintes. Ce problème est NP-complet lorsque  $n$  métriques additives et  $m$  métriques multiplicatives ( $n + m \geq 2$ ) sont présentes. Nous avons considérés plusieurs combinaisons de deux, trois et quatre métriques. Pour une métrique concave et autre additive ou multiplicative, un algorithme linéaire qui choisit un plus court chemin parmi plusieurs chemins de même largeur est proposé. Cet algorithme trouve des chemins sans boucles. Pour les autres combinaisons, un algorithme quasi-polynomial ( $O(m^2 \log^4 m)$ ) où  $m$  est le nombre des arcs) basé sur la relaxation lagrangienne est proposé. Il présente des performances proches de celles des algorithmes exhaustifs qui trouvent des solutions exactes en un temps qui peut être exponentiel. Cet algorithme est utilisé par la suite dans notre protocole de routage avec QoS : le QOLSR.

Le problème de la QoS dans les réseaux ad hoc est considéré. Un modèle de QoS (EQMM) est proposé. Il assure une différenciation de services dans les réseaux ad hoc. Il prend en considération l'ensemble des problèmes qu'une transmission peut subir. Les mécanismes de ce modèle sont : un protocole de routage avec QoS, une estimation des métriques, un contrôle d'admission et de réservation et un contrôle de congestion.

Le protocole QOLSR est un protocole proactif basé sur les principes du protocole OLSR pour offrir la QoS dans les réseaux MANETs. Les messages de contrôle *Hello* et *TC* permettent respectivement à chaque nœud d'acquiescer la topologie ainsi les conditions de QoS locales et globales. Les nœuds relais peuvent être des MPRs classiques, MPRs-1 ou MPRs-2. Nous avons montré que l'utilisation des nœuds MPRs-2 permet de trouver des chemins dans la topologie partielle de même performances que ceux trouver dans la topologie entière. Le nombre de nœuds MPRs-2 peut être plus large que celui des MPRs. Cependant, ils se convergent dans un graphe dense. Les messages *TC* sont générés à chaque changement dans la topologie locale (lien bidirectionnel) ainsi les valeurs de QoS (bande passante et délai) sur les liens. Avec `BANDWIDTH_THRESHOLD = 20%`, le protocole QOLSR réalise le meilleur taux de délivrance mais par contre l'overhead et le délai de bout-en-bout augmentent. Le calcul des routes se fait selon le type de flux. Pour un flux best-effort, la route est calculée en utilisant l'algorithme des chemins shortest-widest ; tandis que les routes des flux QoS sont calculées par l'algorithme de la relaxation lagrangienne. Par simulation, QOLSR a montré ses performances par rapport au protocole OLSR. Le taux de délivrance des flux est très élevé.

L'estimation des métriques est un élément important pour déterminer les valeurs de QoS sur les liens. Il est plus facile de calculer correctement une métrique en se basant sur la couche MAC. Le temps de séjour, le nombre de paquets dans le buffer MAC, la probabilité de perte, le taux de repos dans le canal, etc, sont des paramètres utilisés dans les modèles théoriques développés pour la bande passante et le délai. Les cartes sans fil IEEE 802.11 actuelles ne fournissent aucun de ces paramètres. Dans ce contexte, nous avons proposé deux méthodes expérimentales. Le délai est calculé à la base des messages *Hello* échangés. En effet, les messages *Hello* envoyés respectivement par un nœud et son voisin représentent la transmission de données et la réception de l'ACK. La bande passante doit considérer les transmissions dans le voisinage. Un modèle d'interférence de distance-H pour les transmissions unicast est utilisé pour construire le graphe de conflit des liens. Un autre de distance-K pour les transmissions broadcast est utilisé pour construire le graphe de conflit des nœuds. Les deux graphes combinés permettent de calculer la bande passante disponible sur un lien. Le système linéaire des équations généré par les cliques des deux graphes est corrigé par un facteur de correction  $\beta$ . La meilleure valeur de la distance H obtenue par simulation est 2.

Le contrôle d'admission et de réservation permet de limiter l'apparition des congestions et maintenir les conditions de QoS sur la route sélectionnée. Une signalisation à base de sonde est dédiée pour ce mécanisme. Son fonctionnement consiste à vérifier, admettre, rejeter, réserver, main-

tenir et libérer les ressources sur les chemins trouvés par QOLSR pour les flux QoS.

Les congestions dans MANET peuvent apparaître fréquemment avec la mobilité. Elles peuvent être contournées par un mécanisme du contrôle de congestion. Ce dernier consiste en deux algorithmes : la gestion des files et l'ordonnancement. Ce mécanisme est en cours de développement.

## 7.2 Perspectives

Dans la continuité du travail présenté, nous pourrions approfondir notre étude afin d'améliorer les résultats obtenus dans le domaine des réseaux 4G et la QoS dans les réseaux ad hoc.

Étudier l'impact de la mobilité sur la QoS dans un contexte 4G et, plus particulièrement, l'impact du handover vertical sur la QoS est un axe de recherche important. Ceci permet la continuité de service entre les différentes technologies sans fil d'une part et les réseaux sans fil et fixe d'autre part. On trouve parmi les applications, la téléphonie. Il s'agit de passer automatiquement du réseau mobile GSM à un réseau de téléphonie sur l'Internet sans dégradation de qualité lorsque l'on arrive chez soi, dans la zone de connexion Wi-Fi. Le système fonctionne également dans l'autre sens : une personne est appelée sur son téléphone mobile avec un préfixe 06. Si on détecte qu'elle est chez elle dans la zone d'interaction de son réseau Wi-Fi, l'appel est rerouté sur sa connexion de téléphonie sur ADSL (préfixe 08XX). La même chose si on utilise le service de la visiophonie avec la technologie de l'UMTS.

Pour la qualité de service dans les réseaux ad hoc, plusieurs tâches restent à faire :

- Concevoir et analyser des protocoles pour la gestion des files d'attente et l'ordonnancement afin de contourner rapidement les régions de congestions.
- Simuler le fonctionnement global de modèle de QoS (EQMM).
- Étudier le problème de passage à l'échelle de EQMM.
- Mettre en œuvre et tester le modèle EQMM dans le cadre de projet SAFARI.

Le protocole QOLSR était le but de plusieurs publications et drafts IETF. Les prochaines versions de notre draft IETF vont permettre d'introduire les multiples interfaces ainsi l'interaction avec des mobiles qui mettent en œuvre uniquement OLSR. Une autre version draft IETF de modèle EQMM sera prochainement disponible. Malgré les progrès réalisés, la vision de la QoS dans MANET est toujours ambiguë. Un groupe de discussion est en cours de construction afin d'organiser et unifier l'ensemble des travaux. Sa première étape consiste à définir la QoS dans MANET ainsi que les problèmes et les objectifs attendus.

Nous avons commencé un autre axe théorique concernant la modélisation de la QoS dans la théorie des graphes aléatoires. Un graphe aléatoire classique ne peut pas modéliser un réseau ad hoc puisque les probabilités de génération des liens sont indépendantes et peuvent être même fixes. La probabilité d'existence d'un lien doit être variable entre chaque paire de nœuds selon le modèle de propagation des ondes radio et la présence des obstacles. Nous avons déjà fixé un modèle théorique pour générer des réseaux ad hoc. Maintenant, ce modèle nécessite un nouvel outil mathématique pour exprimer les propriétés comme le nombre de voisins, le diamètre, le nombre de nœuds relais, etc. Une fois ces caractéristiques sont établies, la QoS peut être analysée mathématiquement.

Le protocole QOLSR est un protocole de routage unicast avec QoS. Introduire la QoS dans les protocoles de routage multicast me semble un domaine de recherche important. La mise en œuvre



d'un protocole de multicast peut s'avérer utile car elle offre un moyen robuste pour joindre des destinataires dont l'adresse n'est pas connue à priori ou qui change régulièrement. Il est important de réduire le nombre de transmissions (et la consommation d'énergie au sein des mobiles) dans un réseau sans fil car la bande passante est limitée. Le multicast doit permettre d'optimiser la gestion du médium radio en évitant les retransmissions superflues de messages et en tirant parti de la caractéristique de diffusion inhérente au médium radio.

## Annexe A

# Spécification des paquets de contrôle de QOLSR

### A.1 Le nouveau format d'un message *Hello*

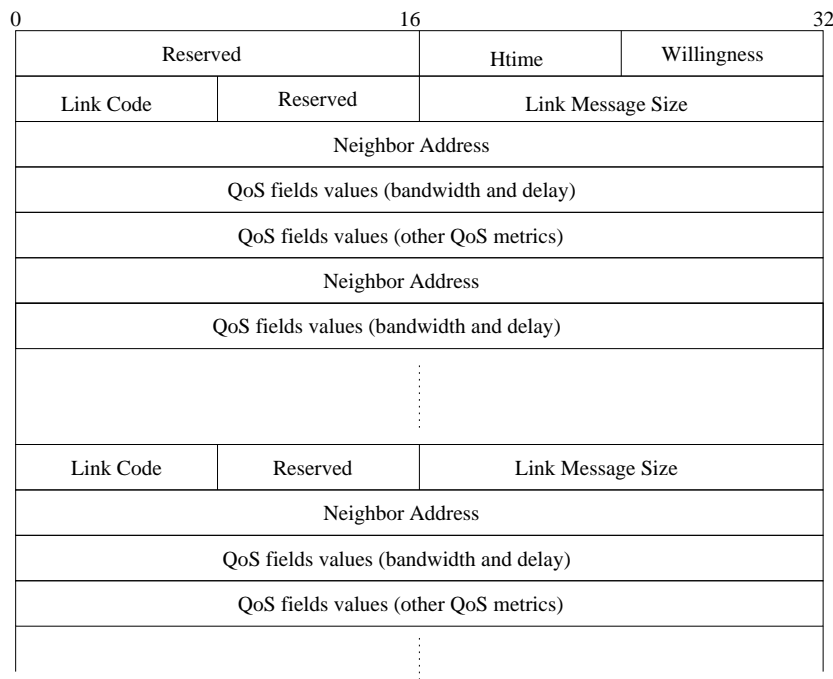


FIG. A.1 – Le format d'un message *Hello*.

Les champs “Reserved”, “Htim”, “Willingness”, “Link Code”, “Link Message Size” et “Neighbor Address” sont décrits dans [73]. Les champs de QoS sont définis selon les scénarios et les besoins de QoS. Comme la bande passante et le délai sont les paramètres de QoS les plus demandées, deux champs sont réservés explicitement et d’une manière permanente. Les autres paramètres peuvent varier d’un scénario à un autre, mais chacun possède un code fixe.

Les champs suivants doivent apparaître dans l’ordre :

- *available bandwidth* : un champ de 24 bits qui donne une mesure en Kbits/seconde. Les

16 bits de poids fort représentent la partie entière et les 8 bits de poids désigne la partie décimale.

- *delay* : un champ de 8 bits qui donne une mesure en milliseconde. Il est codé sous un format à virgule flottante avec une mantisse (4 bits de poids fort) et un exposant (4 bits de poids faible). c’est-à-dire :

$$\text{Delay} = C * (1 + a/16) * 2^b \text{ [en millisecondes]},$$

où  $a$  et  $b$  sont des entiers représentés respectivement par les 4 bits de poids fort et les 4 bits de poids faible du champ délai. La valeur proposée de facteur d’échelle  $c$  est 1/16 millisecondes.

- les 32 bits restants représentent les autres valeurs des paramètres de QoS qui doivent être échangées entre les nœuds voisins pour calculer la valeur finale. La probabilité de perte ( $L_p$ ) est l’un de ces paramètres. Soient  $A$  et  $B$  deux nœuds voisins,  $L_p(A \longleftrightarrow B) = 1 - [(1 - L_p(A \longleftarrow B)) * (1 - L_p(b \longleftarrow A))]$ , et donc la valeur finale de  $L_p$  est calculable en collaboration entre les nœuds voisins. Puisque ces paramètres de QoS ne sont pas figés et peuvent changer d’un scénario à un autre, un format de codage TVL (type/length/Value) est utilisé (section A.3).

## A.2 Le nouveau format d’un message TC

0	16	32
ANSN		Reserved
Multipoint Relay Selector Address		
QoS fields values (bandwidth and delay)		
QoS fields values (other QoS metrics)		
⋮		
Multipoint Relay Selector Address		
QoS fields values (bandwidth and delay)		
QoS fields values (other QoS metrics)		
⋮		

FIG. A.2 – Le format d’un message TC.

Les champs “ANSN”, “Reserved” and “Multipoint Relay Selector Address” sont décrits dans [73]. Comme la bande passante et le délai sont les paramètres de QoS les plus demandées, deux champs sont leurs sont réservés explicitement et d’une manière permanente (voir section A.1). Un format de codage TVL est utilisé pour les paramètres restants.

## A.3 Le codage TVL

La figure A.3 illustre le format d’un codage TVL (type/length/Value).

FIG. A.3 – *Le format de codage TLV.*

Le type et la longueur sont montrés dans la table A.1.

La métrique de QoS	Type	Longueur
Probabilité de perte	0	8 bits
Gigue	1	8 bits
Consommation de puissance	2	8 bits
Coût	3	8 bits
Buffer	4	8 bits
Stabilité	5	8 bits
Sécurité	6	8 bites
⋮	⋮	⋮

TAB. A.1: *Le codage TVL proposé.*



## Annexe B

# Liste de publications

### Chapitre d'ouvrage

- Hakim Badis and Khaldoun Al Agha, *QoS in Wireless Ad hoc Networks*. Advanced Networks and Mobile Computing, Kluwer Academic Publishers, 2004. À paraître.

### Revue internationale avec comité de lecture

- Hakim Badis and Khaldoun Al Agha, *QOLSR, QoS routing for Ad Hoc Wireless Networks Using OLSR*. European Transactions on Telecommunications, Vol. 16, No. 4 : 427-442, 2005.
- Hakim Badis, *An Efficient Optimization of Network Resource Allocations Under Nonlinear Quality of Service Constraints*. IECIE Transactions, Vol. E88, No. 10, October 2005.
- Hakim Badis and Khaldoun Al Agha, *Efficient Vertical Handoffs in Wireless Overlay Networks*. COMPUTING AND INFORMATICS, 2005. À paraître.

### Articles de conférences internationales avec actes et comité de lecture

- Ignacy Gawędzki, Hakim Badis and Khaldoun Al Agha, *Link capacity estimation in QOLSR*. The 2nd OLSR Interop and Workshop, Paris, France, July 2005.
- Hakim Badis and Khaldoun Al Agha, *Routing Bandwidth Guaranteed paths for QoS Flows in Ad Hoc Networks under Interferences Influence*. The First International IEEE CSIT Conference, Algeria, July 2005.
- Hakim Badis, Ines Ben Hamida, Lila Boukhatem and Khaldoun Al Agha, *Integration of Mobile IPv6 and OLSR for Inter-MONET Communications*. The 4th International Med-HOC-Net 2005, France, June 2005.
- Alia Fourati, Hakim Badis and Khaldoun Al Agha, *Security Vulnerabilities Analysis of the OLSR Routing*. ICT'05 : 12th International Conference on Telecommunications, Cape Town, South Africa, May 2005.
- Hakim Badis and Khaldoun Al Agha, *Fast and Efficient Vertical Handoffs in Wireless Overlay Networks*. IEEE PIMRC'04 : Personal Indoor and Mobile Radio Communications,

Barcelona, Spain, September 2004.

- Hakim Badis and Khaldoun Alagha, *An Efficient QOLSR Extension Protocol For QoS in Ad hoc Networks*. IEEE VTC'04-Fall : Vehicular Technology Conference, Los Angeles, USA, September 2004.
- Hakim Badis, Ignacy Gawedzki and Khaldoun Al Agha, *QoS Routing in Ad hoc Networks Using QOLSR with no Need of Explicit Reservation*. IEEE VTC'04-Fall : Vehicular Technology Conference, Los Angeles, USA, September 2004.
- Hakim Badis, Ines Ben Hamida, Lila Boukhatem and Khaldoun Al Agha, *An ad hoc networking for Inter-Monet Using Mobile IPv6 and OLSR*. IEEE VTC'04-Fall : Vehicular Technology Conference, L Angeles, USA, September 2004.
- Hakim Badis and Khaldoun Al Agha, *QOLSR Multi-path Routing for Mobile Ad Hoc Networks Based on Multiple Metrics*. IEEE VTC'04-Spring : Vehicular Technology Conference, Milan, Italy, May 2004.
- Hakim Badis, Anelise Munaretto, Khaldoun Al Agha and Guy pujolle, *Optimal path selection on a link state QoS routing*. IEEE VTC'04-Spring : Vehicular Technology Conference, Milan, Italy, May 2004.
- Hakim Badis and Kahldoun Al Agha, *Reducing vertical handoff latency in wireless overlay networks*. ICMU'04 : International Conference on Mobile Computing and Ubiquitous Networking, Yokosuka, Japan, January 2004.
- Hakim Badis, Anelise Munaretto, Khaldoun Al Agha and Guy pujolle, *QoS for ad hoc networking based on multiple-metric : Bandwidth and delay*. IFIP MWCN'03 : International Workshop On Mobile and Wireless Communications Networks, Singapore, October 2003.
- Hakim Badis and Kahldoun Al Agha, *A distributed algorithm for multiple-metric link state QoS routing problem*. IFIP MWCN'03 : International Workshop On Mobile and Wireless Communications Networks, Singapore, October 2003.
- Hakim Badis and Khaldoun, *An efficient mobility management in wireless overlay networks*. IEEE PIMRC'03 : Personal Indoors and Mobile Radio Communications, Beijing, China, September 2003.
- Hakim Badis and Khaldoun Al Agha, *Scalable model for the simulation of OLSR and Fast-OLSR protocols*. IFIP Med-Hoc-Net'03, Mahdia, Tunisia, June 2003.
- Anelise Munaretto, Hakim Badis, Khaldoun Al Agha, Guy Pujolle, *QoS-enhanced OLSR protocol for mobile Ad Hoc networks*. ANWIRE 1st International Workshop, Glasgow, Scotland, April 2003.
- Anelise Munaretto, Hakim Badis, Khaldoun Al Agha and Guy Pujolle, *A link-state qos routing protocol for ad hoc networks*. IEEE MWCN'02 : International Workshop On Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.

## Articles de conférences nationales avec actes et comité de lecture

- Anelise Munaretto, Hakim Badis, Khaldoun Al Agha, Guy Pujolle, *QOLSR : Routage avec QoS dans OLSR*. AlgoTel 2003, Banyuls-sur-mer, France, INRIA, May 2003.

- Hakim Badis and Khaldoun Al Agha, *Un modèle générique pour la simulation du protocole de routage OLSR*. DNAC'02 : De Nouvelles Architectures pour les Communications, Paris, France, December 2002.

### Drafts de standardisation

- Hakim Badis and Khaldoun Al Agha, *Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR)*. IETF-61 Meeting : Internet Engineering Task Force, draft-badis-manet-qolsr-02.txt, Canada, November 2005.
- Hakim Badis and Khaldoun Al Agha, *Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR)*. IETF-61 Meeting : Internet Engineering Task Force, draft-badis-manet-qolsr-01.txt, Paris, France, August 2005.
- Hakim Badis, Khaldoun Al Agha and Anelise Munaretto, *Quality of Service for Ad hoc Optimized Link State Routing Protocol (QOLSR)*. IETF-61 Meeting : Internet Engineering Task Force, draft-badis-manet-qolsr-00.txt, Washington, DC, USA, November 2004.
- J.P. Jeong, J.-S. Park, K. Mase, Y.-H. Han, H. Badis and J.-M. Orset, *Dynamic Requirements for Ad Hoc IP Address Autoconfiguration*. IETF-61 Meeting : Internet Engineering Task Force, draft-jeong-manet-addr-autoconf-reqts-04.txt, Canada, November 2005.
- J.P. Jeong, J.-S. Park, K. Mase, Y.-H. Han, H. Badis and J.-M. Orset, *Dynamic Requirements for Ad Hoc IP Address Autoconfiguration*. IETF-61 Meeting : Internet Engineering Task Force, draft-jeong-manet-addr-autoconf-reqts-03.txt, Washington, DC, USA, November 2004. Update Version.
- J.P. Jeong, J.-S. Park, K. Mase, Y.-H. Han, H. Badis and J.-M. Orset, *Dynamic Requirements for Ad Hoc IP Address Autoconfiguration*. IETF-60 Meeting : Internet Engineering Task Force, draft-jeong-manet-addr-autoconf-reqts-02.txt, San Diego, USA, August 2004.
- J.P. Jeong, J.-S. Park, K. Mase, Y.-H. Han, H. Badis and J.-M. Orset, *Dynamic Requirements for Ad Hoc IP Address Autoconfiguration*. IETF-60 Meeting : Internet Engineering Task Force, draft-jeong-manet-addr-autoconf-reqts-01.txt, Seoul, Korea, February 2004.
- J.P. Jeong, J.-S. Park, K. Mase, Y.-H. Han, H. Badis and J.-M. Orset, *Dynamic Requirements for Ad Hoc IP Address Autoconfiguration*. IETF-60 Meeting : Internet Engineering Task Force, draft-jeong-manet-addr-autoconf-reqts-00.txt, Minneapolis, USA, July 2003.





# Bibliographie

- [1] “International data corporation.” [Http ://www.idc.com/france](http://www.idc.com/france).
- [2] [http ://www.01net.com/article/236919.html](http://www.01net.com/article/236919.html).
- [3] “IETF, Mobile Ad hoc Network (manet).” [Http ://www.ietf.org/html.charters/manet-charter.html](http://www.ietf.org/html.charters/manet-charter.html).
- [4] “The internet engineering task force.” [Http ://www.ietf.org](http://www.ietf.org).
- [5] Z. Wang and J. Crowcroft, “Quality of service routing for supporting multimedia applications,” in *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1228–1234, September 1996.
- [6] [http ://www.telecom.gouv.fr/rnrt/rnrt/projets/res\\_02\\_04.htm](http://www.telecom.gouv.fr/rnrt/rnrt/projets/res_02_04.htm).
- [7] K. Al Agha, G. Pujolle and G. Vivier, *Réseaux de mobiles & Réseaux sans fil*. No. ISBN 2-212-11018-9, Eyrolles, 2001.
- [8] C. Perkins, “IP Mobility Support for IPv4,” January 2002.
- [9] C. Perkins, “Mobile IP Design Principles and Practices,” January 1998.
- [10] “Third generation partnership project 2.” [Http ://www.3gpp2.org](http://www.3gpp2.org).
- [11] M. Mouly and M. Pautet, “The GSM System for Mobile Communications,” June 1992.
- [12] Javier Sanchez and Mamadou Thioune, *UMTS : services, architecture et WCDMA*. Lavoisier, 2001.
- [13] J. S. DaSilva, B. Arroyo, B. Barani and D. Ikonomou, “European third generation mobile systems,” in *IEEE Communications Magazine*, vol. 34, pp. 68–83, Octobre 1995.
- [14] K. Al Agha, *Évoluer vers la 4<sup>e</sup> génération*. Habilitation à diriger des recherches, université de Paris-Sud XI, October 3 2002.
- [15] R. Caceres and V. N. Padmanabhan, “Fast and scalable Handoffs for wireless Internet-networks,” in *ACM International Conference on Mobile Computing and Networking*, (New York, USA), pp. 56–66, November 1996.
- [16] C. Perkins and D. Johnson, “Route Optimization in Mobile IP,” draft-ietf-mobileip-optim-09.text, IETF : The Internet Engineering Task Force, [http ://www.ietf.org](http://www.ietf.org), Febraury 2000.
- [17] E. Gustafsson, A. Jonsson and C. Perkins, “Mobile IP Regional Registration,” draft-ietf-mobileip-reg-tunnel-04.text, IETF : The Internet Engineering Task Force, [http ://www.ietf.org](http://www.ietf.org), January 2001.
- [18] A.G. Valko , “A New Approach to Internet Host Mobility,” in *ACM Computer Communication Review*, January 1999.

- [19] A. Campbell, J. Gomez, C. Wan, Z. Turanyi and A.G. Valko, "Cellular IP," draft-valko-cellularip-00.text, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, January 2000.
- [20] A. Campbell, J. Gomez, S. Kim, A.G. Valko, C.Y. Wan and Z. Turany, "Design, Implementation and Evaluation of Cellular IP," in *IEEE Personal Communications, Special Issue on IP-based Mobile Telecommunications Networks*, no. 42-49, pp. 56–66, August 2000.
- [21] R. Ramjee, T.L. Porta, S. Thuel, K. Varadhan and L. Salgarelli, "IP micro-mobility support using HAWAII," draft-ietf-mobileip-hawaii-00-txt, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, June 1999.
- [22] R. Ramjee, T.L. Porta, S. Thuel, K. Varadhan and S.Y. Wang, "HAWAII : a Domain-based Approach for supporting Mobility in Wide-area Wireless Networks," in *IEEE International Conference on Network Protocols*, (Toronto), November 1999.
- [23] S. Das, A. Misra, P. Agrawal and S.k. Das, "TeleMIP : Telecommunication Enhanced Mobile IP Architecture for Fast Intra-Domain Mobility," in *IEEE Personal Communications*, pp. 50–58, August 2000.
- [24] "Bluetooth specifications." <Http://www.bluetooth.com>.
- [25] ANSI/IEEE std 802.11, "Wireless LAN Medium Access Control (MAC) and physical Layer Specifications," tech. rep., 1999.
- [26] B. O'Hara and A. Patrik, "IEEE802.11 Handbook a Designer's Companion," in *Standards Information Network*, IEEE Press, 1999.
- [27] ETSI STC-RES10 Committee, "Radio Equipment and Systems : High Performance Radio Local Area Network Type 1," in *Functional specifications*, June 1996.
- [28] J. Sun and J. Sauvola, "Mobility and mobility management : a conceptual framework," pp. 205–210, August 2002.
- [29] F. Di Cola, P.M.L. Chan, R.E. Sheriff and Y.F. Hu, "Handover and QoS Support in Multi-Segment Broadband Networks," September 2000.
- [30] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioglu and W. Wang, "Mobility Management in Next Generation Wireless Systems," in *IEEE Proceedings Journal*, vol. 87, pp. 1347–1385, 1999.
- [31] N.D. Tripathi, J.H. Reed and H.F. VanLandingham, "Handoff in Cellular Systems," in *IEEE Personal Communications. Magazine*, pp. 26–37, December 1998.
- [32] 3GPP TS 23.334 V6.2.0, "3GPP system to Wireless Local Area Network (WLAN) interworking ; System description (Release 6)," tech. rep., September 2004.
- [33] 3GPP TR 22.934 V6.2.0, "Feasibility study on 3GPP system to Wireless Local Area Network (WLAN) interworking (Release 6)," tech. rep., September 2003.
- [34] S.-L. Tsao, C.-C. Lin, "VGSN : A Gateway Approach to Interconnect UMTS/WLAN Networks," in *13th IEEE International Symposium on Personal, Indoor, Mobile Radio Communications*, vol. 1, pp. 275– 279, September 2002.
- [35] V. Varma, S. Ramesh, K. D. Wong, M. Barton, G. Hayward and J. Friedhoffer, "Mobility Management in Integrated UMTS/WLAN Networks," in *IEEE ICC*, (Alaska, USA), May 2003.

- [36] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP : Session Initiation Protocol," RFC 3261, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, June 2002.
- [37] N. Vulic, I. Niemegeers and S. Heemstra de Groot, "Architectural options for the WLAN integration at the UMTS radio access level," in *IEEE Vehicular Technology Conference Spring*, (Milan, Italy), May 2004.
- [38] S.-L. Tsao and C.-C. Lin, "Design and evaluation of UMTS/WLAN interworking strategies," in *IEEE Vehicular Technology Conference Fall*, (Canada), September 2002.
- [39] M. Stemm, P. Gauthier, D. Harada and R. H. Katz, "Reducing Power Consumption of Network Interfaces in Hand-Held Devices," in *MoMuC-3 : 3rd Workshop on Mobile Multimedia Communications*, September 1996.
- [40] "AT&T WaveLAN : PC/AT card Installation and Operation," tech. rep., AT&T manual, 1994.
- [41] <http://www.opnet.com>.
- [42] H. Badis and K. Al Agha, "QOLSR, QoS routing for Ad Hoc Wireless Networks Using OLSR," in *European Transactions on Telecommunications*, vol. 15, 2005.
- [43] I. Microelectronics, "IBM Infrared Wireless LAN Adapter Technical Reference," tech. rep., Toronto Lab, 1995.
- [44] Mark Stemm and Randy H. Katz, "Vertical handoffs in Wireless overlay networks," in *Mobile Networks and Applications*, vol. 3, pp. 70–80, 1999.
- [45] E. Duros and W. Dabbous, "Supporting unidirectional Paths in the Internet," draft-ietf-udlr-general-00.txt, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, June 1996.
- [46] R. Widyono, "The design and evaluation of routing algorithms for realtime channels," technical report tr-94-024, University of California at Berkeley, June 1994.
- [47] R. Hassin, "Approximation schemes for the restricted shortest path problem," in *Mathematics of Operations Research*, vol. 17, pp. 36–42, February 1992.
- [48] D.H. Lorenz and A. Orda, "QoS routing in networks with uncertain parameters," in *IEEE/ACM Transactions on Networking*, vol. 6, pp. 768–778, December 1998.
- [49] D. Raz and Y. Shavitt, "Optimal Partition of QoS requirements with Discrete Cost Functions," in *IEEE J. on Selected Areas in Communications*, vol. 18, pp. 2593–2602, December 2000.
- [50] Dean H. Lorenz, Ariel Orda, Danny Raz and Yuval Shavitt, "Efficient QoS Partition and Routing of Unicast and Multicast," in *IWQoS 2000*, (Pittsburgh, PA, USA), pp. 75–83, June 2000.
- [51] W.C.Lee and et al, "Multi-Criteria Routing subject to Resource and Performance Constraints," tech. rep., ATM Forum 94-0280, March 1994.
- [52] K. Ishida, K. Amano and N. Kannari, "A delay-constrained least-cost path routing protocol and the synthesis method," in *the Fifth IEEE International Conference on Real-Time Computing Systems and Applications*, pp. 58–65, October 1998.
- [53] Hussein F. Salama and Douglas S. Reeves and Yannis Viniotis, "A Distributed Algorithm for Delay-Constrained Unicast Routing," in *INFOCOM*, (Japan), pp. 84–91, April 1997.

- [54] C. Pornavalai, G. Chakraborty and N. Shiratori, "Routing with multiple QoS requirements for supporting multimedia applications," in *Telecommunication Systems*, vol. 9, pp. 357–373, 1998.
- [55] S. Chen and K. Nahrstedt, "On finding multiconstrained paths," in *IEEE International Conference Communications (ICC98)*, (Atlanta, GA), pp. 874–879, June 1998.
- [56] E. W. Dijkstra, "A note on two problems in connection with graphs," in *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [57] R. Bellman, "On a Routing Problem," in *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.
- [58] S. Chen and K. Nahrstedt, "On finding multiconstrained paths," in *Technical Report UIUCDCS-R-97-2006*, (Department of Computer Science, University of Illinois at Urbana-Champaign), August 1997.
- [59] J. M. Jaffe, "Algorithms for finding paths with multiple constraints," in *Networks*, pp. 95–116, 1984.
- [60] H. De Neve and P. Van Mieghem, "A multiple quality of service routing algorithm for PNNI," in *IEEE ATM Workshop*, (George mason university, USA), pp. 324–328, May 1998.
- [61] T. Korkmaz, M. Krunz and S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints," in *SIGMETRICS '00 : Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, vol. 1, (Santa Clara, California, USA), pp. 318–327, June 2000.
- [62] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," in *J. ACM*, vol. 34, (New York, USA), pp. 596–615, ACM Press, 1987.
- [63] Q. Ma and P. Steenkiste, "Quality of service Routing for traffic with Performance Guarantees," in *IFIP Fifth International Workshop on Quality of Service*, (New York, USA), pp. 115–126, May 1997.
- [64] L.H. Andrew and A.N. Kusuma, "Generalized analysis of a QoS-aware routing algorithm," in *IEEE GLOBECOM 1998*, vol. 1, (New Sydney, Australia), pp. 1–6, November 1998.
- [65] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda and T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions," RFC 2676, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, August 1999.
- [66] A. M. Geoffrion, "Lagrangian relaxation for integer programming," in *Mathematical Programming, Study 2*, vol. 1, 1974.
- [67] M. L. Fisher, W. D. Northup and J. F. Shapiro, "Using duality to solve discrete optimization problems : theory and computational experience," in *Mathematical Programming, Study 3*, vol. 1, pp. 56–94, 1975.
- [68] M. L. Fisher, "The lagrangian relaxation method for solving integer programming problems," in *Management Science*, vol. 27, pp. 56–94, January 1981.
- [69] S. Walukiewicz, "Integer Programming," in *Kluwer Academic Publishers, Noewell, MA.*, 1991.
- [70] T. Radzik, "Fractional combinatorial optimization," in *Handbook of Combinatorial Optimization* (D. Du and P. Pardalos, eds.), vol. 1, pp. 95–116, Kluwer Academic Publishers, December 1998.

- [71] A. Fei and G. Pei and R. Liu and L. Zhang, "Measurements on delay and hop-count of the Internet," in *IEEE GLOBECOM98 - Internet Mini-Conference*, (Sydney, Australia), November 1998.
- [72] G. Chelius and E. Fleury, "ANANAS : A New Ad hoc Network Architectural Scheme," Rapport de Recherche 4354, INRIA, 2002.
- [73] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, October 2003.
- [74] J.P. Macker and M.S. Corson, "Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, January 1999.
- [75] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *ACM SIGCOMM'94*, (London, UK), August 1994.
- [76] J. Malkin, "Routing Information Protocol (RIP) V2," RFC 1388, IETF : The Internet Engineering Task Force, January 1993.
- [77] J. Moy, "Open Shortest Path First (OSPF) V2," RFC 2328, IETF : The Internet Engineering Task Force, January 1998.
- [78] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 2561, IETF : The Internet Engineering Task Force, July 2003.
- [79] D.B. Johnson, D.A. Maltz and Y. Hu, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks (DSR)," draft-ietf-manet-dsr-10.txt, IETF : The Internet Engineering Task Force, July 2004.
- [80] Z.J. Haas and M.R. Pearlman, "The Zone Routing Protocol for Mobile Ad hoc Networks," draft-ietf-manet-zone-zrp-02.txt, IETF : The Internet Engineering Task Force, July 2000.
- [81] M. Jiang, J. Li and Y.C. Tay, "Cluster Based Routing Protocol (CBRP)," draft-ietf-manet-cbrp-spec-01.txt, IETF : The Internet Engineering Task Force, July 1999.
- [82] R. Ogier, F. Templin and M. Lewis, "Topology Dissemination Based on Reserved-Path Forwarding (TBRPF)," RFC 3684, IETF : The Internet Engineering Task Force, February 2004.
- [83] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laoutiti, T. Clausen and L. Viennot, "Optimized Link State Routing Protocol," in *IEEE INMIC*, (Pakistan), December 2001.
- [84] A. Qayyum, A. Laoutiti and L. Viennot, "Multipoint relaying technique for flooding broadcast messages in mobile wireless networks," in *HICSS : Hawaii International Conference on System Sciences*, (Hawaii, USA), January 2002.
- [85] T. Clausen, P. Minet and C. Perkins, "Multipoint Relay Flooding for Manets," draft-perkins-manet-mprf-00.txt, IETF : The Internet Engineering Task Force, February 2004.
- [86] A. Qayyum, L. Viennot and A. Laoutiti, "Multipoint Relaying : An Efficient Technique for Flooding in Mobile Wireless Networks," Research Report 3898, INRIA, February 2000.
- [87] E.W. DIJKSTRA, "A Note en two Problems in Connection with Graphs," in *Num. math*, vol. 1, pp. 269–271, 1959.
- [88] D. Zeinalipour, S. Aristeidou and S. Kazeli, "IP Quality of Services," in *Report*, ([http://www.cs.ucr.edu/csyiazti/downloads/papers/ip qos/papers/ip-qos.pdf](http://www.cs.ucr.edu/csyiazti/downloads/papers/ip%20qos/papers/ip-qos.pdf)), 1999.

- [89] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol, version 1 functional specification," RFC 2205, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, September 1997.
- [90] R. Braden and L. Zhang, "Resource ReSerVation Protocol, version 1 Message Processing Rules," RFC 2209, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, September 1997.
- [91] D. Awduche, L. Berger, T. Li, V. Srinivasan and G. Swallow, "RSVPTE : Extensions to RSVP for LSP Tunnels," RFC 3209, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, December 2001.
- [92] D. Katz, D. Yeung and K. Kompella, "Traffic Engineering Extensions to OSPF version 2," internet draft, draft-katz-yeung-ospf-traffic-09.txt, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, October 2002.
- [93] B. Davie and Y. Rekhter, "MPLS : technology and applications," Morgan Kaufmann, 2000.
- [94] H. Xiao, W. K.G. Seah, A. Lo and K. C. Chua, "A Flexible Quality of Service Model for Mobile Ad-Hoc Networks," in *IEEE Vehicular Technology Conference*, (Tokyo, Japan), pp. 445–449, May 2000.
- [95] J. Heinanen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHP Group," RFC 2597, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, June 1999.
- [96] G. Ahn, A. T. Campbell, A. Veres and L. Sun, "SWAN : Service Differentiation in Stateless Wireless Ad Hoc Networks," in *IEEE INFOCOM02*, vol. 1, (New york, USA), pp. 457–466, May 2002.
- [97] K. Chen, S. H. Shah and K. Nahrstedt, "Cross Layer Design for Data Accessibility in Mobile Ad Hoc Networks," in *Wireless Communications*, vol. 21, (New york, USA), pp. 49–75, 2002.
- [98] S.B. Lee, A. Gahng-Seop, X. Zhang and A.T. Campbell, "INSIGNIA : An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks," in *Parallel and Distributed Computing (Academic Press), Special issue on Wireless and Mobile Computing and Communications*, vol. 60, pp. 374–406, April 2000.
- [99] C. Chaudet and I. Guérin Lassous, "BRuIT : Bandwidth Reservation under InTerferences influence," in *European Wireless*, pp. 466–472, 2002.
- [100] R. Sivakumar, P. Sinha and V. Bharghavan, "CEDAR : a Core-Extraction Distributed Ad hoc Routing algorithm," in *IEEE INFOCOM99*, vol. 1, (New york, USA), pp. 202–209, March 1999.
- [101] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," in *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1488–1504, 1999.
- [102] C. E. Perkins, E. M. Royer and S. R. Das, "Quality of service for ad hoc on-demand distance vector routing," Internet Draft, draft-ietf-manet-aodvqos-00.txt, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, July 2000.
- [103] M. R. Gary and D. S. Johnson, "Computers and intractability," in *A guide to the theory of NP-completeness*, W. H. Freeman, 1979.
- [104] T. Cormen, C. Leireson and R. Rivest, "Introduction to Algorithms," in *MIT press*, 1990.
- [105] A. Qayyum, "Analysis and evaluation of channel access schemes and routing protocols for wireless networks," in *PHD thesis*, (University of Paris XI, France), 2000.

- [106] A. Laoutit, A. Qayyum and L. Viennot, "Multipoint Relaying : An efficient Technique for flooding in mobile wireless networks," in *Rapport de Recherche, RR-3898*, (INRIA, France), March 2000.
- [107] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," in *IEEE Journal on Selected Areas in Communications*, pp. 535–547, March 2000.
- [108] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," in *IEEE transactions on Information Theory*, vol. 34, pp. 388–404, March 2000.
- [109] M. Grossglauser, D. Tse, "Mobility Increases the Capacity of Ad hoc Wireless Networks," in *IEEE/ACM Transactions on Networking*, vol. 10, pp. 477–486, 2002.
- [110] A. El Gamal, J. Mammen, B. Prabhakar and D. Shah, "Throughput-Delay Trade-off in Wireless Networks," in *IEEE INFOCOM04*, (Hong Kong), March 2004.
- [111] V. Jacobson, R. Braden and D. Borman, "TCP Extensions for High Performance," RFC 1323, IETF : The Internet Engineering Task Force, <http://www.ietf.org>, May 1992.
- [112] K. Romer, "Time synchronization in ad hoc networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing*, (Long Beach, CA, USA), pp. 173–182, ACM Press, October 2001.
- [113] Z. Jia, R. Gupta, J. Walerand and P. Varaiya, "Bandwidth Guaranteed routing for Ad hoc Networks with Interference Consideration," in *IEEE ISCC05*, vol. 13, (SPAIN), pp. 03–09, June 2005.
- [114] L. Lovasz, "A characterization of Perfect Graph," in *Journal of Combinatorial Theory*, vol. 13 of *serie B*, pp. 95–98, October 1978.
- [115] A. Graf, M. Stumpf and G. Weissenfls, "On coloring Unit Disk Graphs," in *Algorithmica*, vol. 20, pp. 277–293, 1998.
- [116] S. Skiena, "Maximum Cliques," in *Implementing Discrete Mathematics : Combinatorics and Graph Theory with Mathematica*, pp. 215–218, 1990.